Dr. Dobb's Journal

SOFTWARE TOOLS FOR ADVANCED PROGRAMMERS

#104 JUNE 1985 $2.95 (3.95 CANADA)

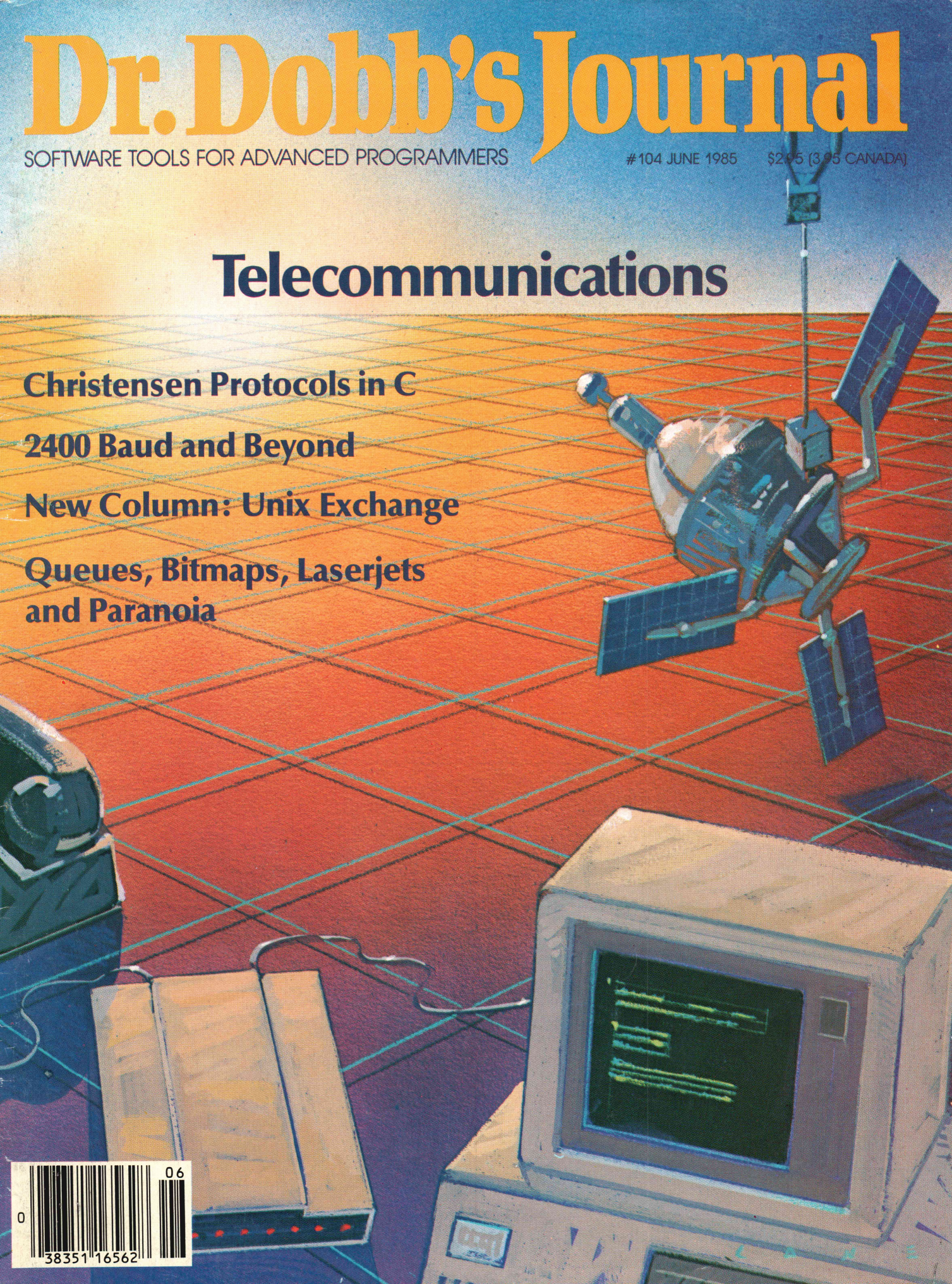
Telecommunications

Christensen Protocols in C

2400 Baud and Beyond

New Column: Unix Exchange

Queues, Bitmaps, Laserjets and Paranoia



0 38351 16562 06

Software development isn't a mountainous task once you eliminate the high C errors.

When you can find and fix bugs at the earliest possible moment, creating software stops being such an uphill grind.

And the Smart/C Environment makes it possible. It's a complete, fully-integrated development environment for C that saves you from the creativity-inhibiting cycle of edit, compile, re-edit, re-compile, link, load, test, re-edit, re-compile, etc., ad infinitum. Smart/C puts the fun back in programming, because you spend your time creating... not waiting.

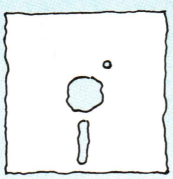
Here's why. Syntax errors are eliminated automatically as code is entered. Smart/C's highly integrated editor and interpreter allow you to interpret your program at any time in the creation process, so logic errors can be ferreted out as soon as the algorithm exists—long before any compile, link, or load.

The complete integration of the editor and interpreter means you can stop anywhere in the interpret cycle, edit, and then go right back into the interpreter exactly where you left off. Not only that, the screen-oriented user interface lets you see all operations, even interpretation, right on the listing of the code.

Circle **no. 4** on reader service card.

And to make maintenance programming easier, Smart/C's Migrator allows existing C code produced with any editor to be modified and run within the Smart/C Environment.

All of which makes Smart/C an excellent tool. It's flexible, non-restrictive, and lets you create elegant, readable, error-free programs that you can watch run with a great feeling of satisfaction.



Smart/C™

Free Demo Disk!

To fully appreciate Smart/C, you have to see it in action. For your free IBM PC MS-DOS demo disk, call us. Or write us on your company letterhead.
AGS Computers, Inc., Advanced Products Division, 1139 Spruce Drive, Mountainside, NJ 07092.
800-AGS-1313. In NJ, 201-654-4321.

Smart/C Features

The Smart/C Environment
- ☐ Fully integrated editor and interpreter
- ☐ Only one load brings them both in
- ☐ One command set
- ☐ Move between one another at will

Syntax Directed Editor
- ☐ vi-like command set
- ☐ Automatically provides formats for blocks, *for*, *case* and *if* statements

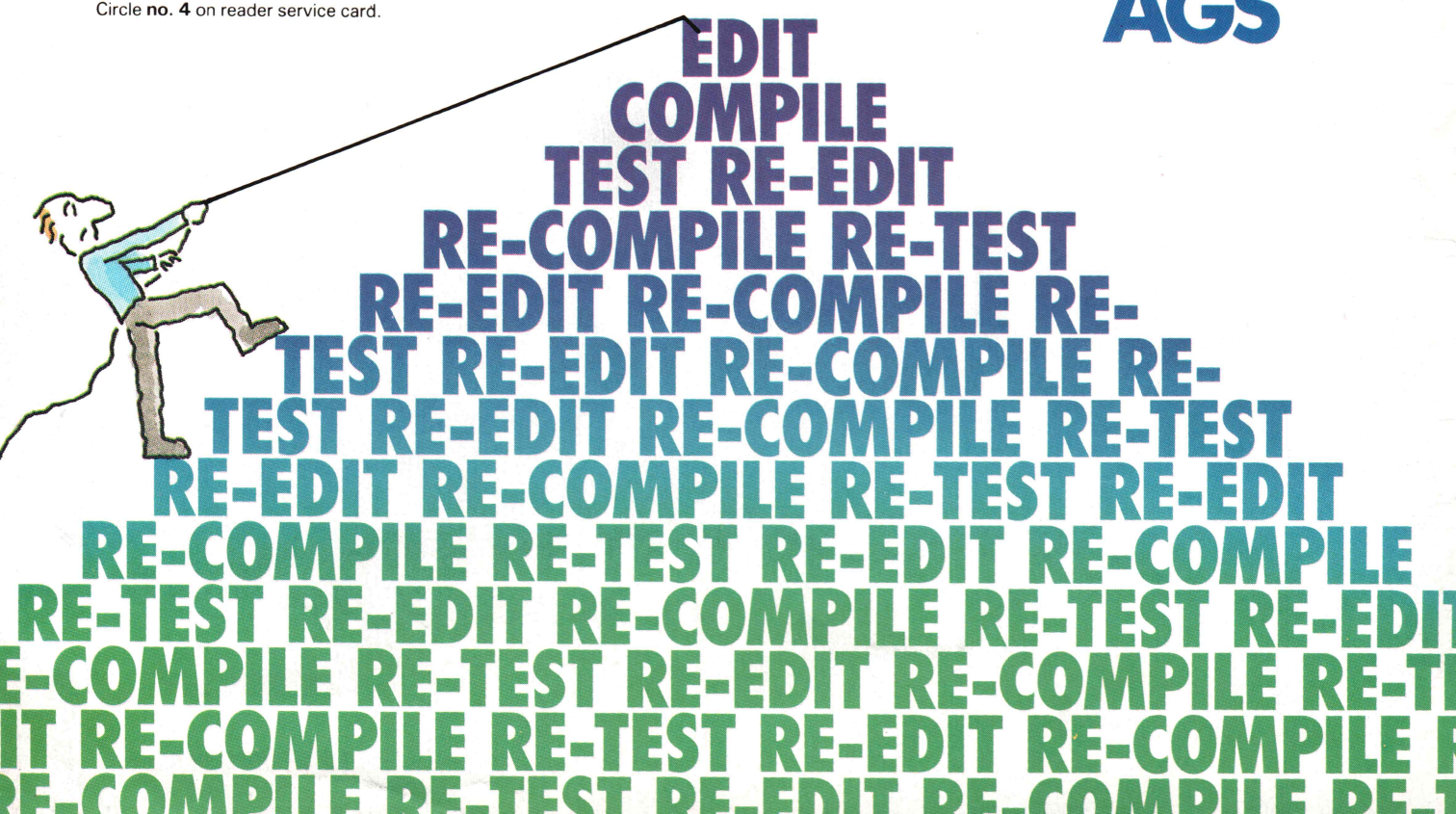
Interpreter
- ☐ Current module can call external modules during interpretation
- ☐ Has Include capability
- ☐ Totally precompilation—no incremental compile
- ☐ Can interpret partially defined files allowing for rapid prototyping
- ☐ Variable speed of interpretation
- ☐ Multiple windows with user-defined sizes

The Smart/C Migrator
- ☐ Allows C code produced with any editor to be interpreted by Smart/C
- ☐ Reformats for readability

Smart/C has been ported to UNIX™ System V Release 2, Berkeley 4.2, Xenix,™ and MS-DOS. Versions run on 8086- and 68000-based machines, as well as proprietary architectures. Smart/C runs on PCs, micros, supermicros, minis, and even mainframes.

Trademarks—Smart/C: AGS Computers, Inc.; UNIX: AT&T Bell Labs; Xenix and MS-DOS: Microsoft Corp; IBM PC: IBM Corp.

AGS



Professional Pascal™

Smaller & Faster code Plus 22 reasons for not using Microsoft® Pascal

The List:

Compile Time
Symbol table space limited only by available memory
Cross referencer
ANSI–standard checking
Option to search a list of directions to find any input file
Source interleaved with Object in Object listing
Full Intel OMF Line/Symbol/Type records for debugger
Run Time
Code quality
Five models of memory on the 8086: small, medium, compact, big and large
Code produced suitable for ROM
Expressions
Long/short integer mixes
Set constant expressions
Set elements in the range $-2^{31}..2^{31}-1$ or 0..256
Type casts
// = Concat for concatenation of all strings
Statements
Iterators (for data abstraction from CLU)
Assignment to string variables of differing lengths
Declarations
Otherwise (or else) in record declarations
Intrinsics
ReadIn(X:M)
Inc, Dec, Incl, Decl, from Modula–2
Data Types
Comparison (for equality) of record or array types
The value -32768 allowed as an integer
Maximum size of a data structure on the 8086

Information on Microsoft Pascal obtained from the 1984 release 3.2. printing of the MS manual.

Microsoft has a problem. The 22 features listed above are missing from MS Pascal.

Microtec Research has the answer. Professional Pascal.

Professional Pascal is more than just powerful features. It generates the most efficient code you've ever seen for 8086/8087/8088/186/188/286/287 in 5 memory models.

Actually 35 positive differences exist in our favor.

In a direct comparison – there isn't any!

*We're **Functional** and **Fast** and **Serious** about our products. We've been providing flexible and economical solutions for software developers since 1974.*

*Beginning with product concept, through development, quality assurance, and post-sales support – **Quality, Compatibility** and **Service** are the differences which set Microtec Research apart from others.*

If you're a serious software developer, shopping for software development tools, call or write today for more information. ***800-551-5554*** *In CA call* ***(408) 733-2919***

No matter what Pascal or chip you choose, recompile with the fully compatible, across all implementations, Professional Pascal.

We were never accused of humility – only solid software development tools.

MICROTEC® RESEARCH

3930 Freedom Circle, Suite 101, Santa Clara, CA 95054 Mailing Address: P.O. Box 60337, Sunnyvale, CA 94088

Circle **no. 38** on reader service card.

Professional Pascal is a trademark of MetaWare, Inc. *Microtec is a registered trademark of Microtec Research, Inc.* *Microsoft is a registered trademark of Microsoft Corporation.*

SuperKey
TOP SECRET
ROBERT
'85 TINNEY

Borland Does It Again: SuperKey $69.95

Sure, ProKey™ is a nice little program. But when the people who brought you Turbo Pascal and SideKick get serious about keyboard enhancers, you can expect the impossible . . . and we deliver.

	SuperKey	ProKey
ALL FEATURES RESIDENT IN RAM AT ALL TIMES	YES	NO
RESIDENT PULL-DOWN MACRO EDITOR	YES	NO
RESIDENT FILE ENCRYPTION	YES	NO
PROKEY COMPATIBILITY	YES	YES
DISPLAY PROTECTION	YES	NO
ABILITY TO IMPORT DATA FROM SCREEN	YES	NO
PULL-DOWN MENU USER INTERFACE	YES	NO
CONTEXT-SENSITIVE ON-LINE HELP SYSTEM	YES	NO
DISPLAY-ONLY MACRO CREATION	YES	NO
ENTRY AND FORMAT CONTROL IN DATA FIELDS	YES	NO
COMMAND KEYS REDEFINABLE "ON THE FLY"	YES	NO
PRICE	69.95	129.95

Total ProKey compatibility. Every Prokey Macro file may be used by SuperKey *without change* so that you may capitalize on all the precious time you've invested.

Now your PC can keep a secret! SuperKey includes a resident file encryption system that uses your password to encrypt and decrypt files, even while running other programs. Two different encryption modes are offered:

1. Direct overwrite encryption (which leaves the file size unchanged) for complete protection. At no point is a second file that could be reconstructed by an intruder generated. Without your secret password, no one will ever be able to type out your confidential letters again!

2. COM or EXE file encryption which allows you to encrypt a binary file into an ASCII file, transmit it through a phone line as a text file and turn it back again into an executable file on the target machine (only of course if your correspondent knows the secret password!). Now, you will even be able to secretly exchange programs through Public Bulletin Board Systems or services such as CompuServe.

Totally memory resident at all times, gives SuperKey the ability to create, edit, save and even recall new or existing macro files anytime, even while running another program.

Pull down macro editor. Finally, a sensible way to create, edit, change and alter existing macro definitions. Even while using another application, a simple keystroke instantly opens a wordprocessor-like window where you're allowed to see, edit, delete, save and even attach names to an individual macro or file of macros, and much more.

Sorry ProKey !

Superb software at reasonable prices!
There is much more to SuperKey. Maybe the best reason to buy SuperKey is that it is a Borland International Product. Each one of our products is the best in its category. We only believe in absolutely superb software at reasonable prices!

An offer you can't refuse.
Whether you are a ProKey user or you've never used a keyboard enhancer before, your boat has come in. You can get your copy of SuperKey at this irresistible price.

Get your PC a SuperKey today!
SuperKey is available now for your IBM PC, XT, AT, jr. and truly compatible microcomputers.

Circle **no. 11** on reader service card.

NOT COPY-PROTECTED

SuperKey

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133

$69.95

This price includes shipping to all U.S. cities. All foreign orders add $10 per product ordered.

YES! Please rush SuperKey to me. Send me ____ copies.

Name: _____

Shipping Address: _____

City: _____ Zip: _____

State: _____

Telephone: _____

Subtotal _____

(CA res. add 4.20 tax per copy) _____

Amount Enclosed: _____

Payment VISA MC BankDraft Check

Credit Card Exp. Date ___/___

Card # _____

F11

COD's and Purchase Orders WILL NOT be accepted by Borland. California residents: add 6% sales tax. Outside USA: add $10 and make payment by bank draft, payable in US dollars drawn on a US bank.

BORLAND INTERNATIONAL

Software's Newest Direction
4585 Scotts Valley Drive
Scotts Valley, CA 95066
TELEX 172373

IBM is a registered trademark of International Business Machine Corporation. ProKey is a trademark of RoseSoft. SuperKey and SideKick are trademarks of Borland International, Inc. CompuServe is a trademark of CompuServe Corp.

Dr. Dobb's Journal

Editorial
Editor-in-Chief *Michael Swaine*
Editor *Randy Sutherland*
Managing Editor *Frank DeRose*
Technical Editor *Alex Ragen*
Editorial Assistant *Sara Noah*
Contributing Editors *Robert Blum, Dave Cortesi, Ray Duncan, Allen Holub*
Copy Editor *Polly Koch*
Typesetter *Jean Aring*

Production
Design/Production Director *Detta Penna*
Art Director *Shelley Rae Doeden*
Production Assistant *Alida Hinton*
Cover *William Cone*

Circulation
Sub. Fulfillment Mgr. *Stephanie Barber*
Subscription Mgr. *Maureen Snee*
Book Marketing Mgr. *Jane Sharninghouse*
Single Copy Sales Mgr. *Kathleen Boyd*

Administration
Finance Manager *Sandra Dunie*
Business Manager *Betty Trickett*
Accounts Payable Supv. *Mayda Lopez-Quintana*
Accounts Payable Asst. *Denise Giannini*
Billing Coordinator *Laura Di Lazzaro*

Advertising
Advertising Director
Stephen Friedman (415) 424-0600
Advertising Sales
Walter Andrzejewski (617) 567-8361
Lisa Boudreau (415) 424-0600
Beth Dudas (714) 643-9439
Michele Beaty (317) 875-0557
DDJ Classifieds
Alice Abrams (415) 424-0600
Advertising Coordinators
Jay Horvath (415) 424-0600
Alice Abrams (415) 424-0600

M&T Publishing, Inc.
Chairman of the Board *Otmar Weber*
Director *C.F. von Quadt*
President *Laird Foshay*

Entire contents copyright © 1985 by M&T Publishing, Inc. unless otherwise noted on specific articles. All rights reserved.

Dr. Dobb's Journal (USPS 307690) is published monthly by M&T Publishing, Inc., 2464 Embarcadero Way, Palo Alto, CA 94303, (415) 424-0600. Second class postage paid at Palo Alto and at additional entry points.

Address correction requested. Postmaster: Send Form 3579 to *Dr. Dobb's Journal*, 2464 Embarcadero Way, Palo Alto, CA 94303. **ISSN 0278-6508**

Subscription Rates: $25 per year within the United States, $44 for first class to Canada and Mexico, $62 for airmail to other countries. Payment must be in U.S. Dollars, drawn on a U.S. Bank. For subscription problems, call: outside of CA 1-800-321-3333; within CA 1-619-485-6535.

Foreign Distributors: ASCII Publishing, Inc. (Japan), Computer Services (Australia), Computer Store (New Zealand), Computercollectief (Nederland), Homecomputer Vertriebs GMBH (West Germany), International Presse (West Germany), La Nacelle Bookstore (France), McGill's News Agency PTY LTD (Australia), Progresco (France).

People's Computer Company

Dr. Dobb's Journal is published by M&T Publishing, Inc. under license from People's Computer Company, 2682 Bishop Dr., Suite 107, San Ramon, CA 94583, a non-profit, educational corporation.

June 1985
Volume 10, Issue 6

CONTENTS

The View from Mid-Year

These are the articles that generated the most reader response in recent months:

January, "Fatten Your Mac" by Thomas LaFleur and Susan Raab, and "Archiving Files with CP/M 80 and CP/M 86" by Ian Ashdown

February, Excerpts from *Fire in the Valley* by Paul Freiberger and Michael Swaine, and "CP/M Exchange," in which Bob Blum discusses buffered disk I/O

March, Richard Stallman's "GNU Manifesto," and Dave Cortesi's "Tour of Prolog"

April, Allen Holub's general purpose Quicksort routine in "C Chest"

May, "Solid Shape Drawing on the Commodore-64" by Richard Rylander

This month, Dean Gengle, our featured author, asks how we are to maintain democracy in an information age. There is more to telecommunications than baud rates and band widths! Special thanks to Ian Ashdown of byHeart Software and Steven Bellovin of AT&T Bell Laboratories for serving as referees.

Next month we deliver three hardware articles: "Build a Custom PC or Clone," "The Ultimate Parallel Print-Spooler," and the feature "Designing a Real-Time Clock for the S-100 Bus." In "16-Bit Toolbox," Ray Duncan presents MSDOS installable device drivers with an overview of the Unix device drivers which inspired them. Allen Holub, in "C Chest" offers an MSDOS directory utility which includes a multicolumn print utility. "Computer Calisthenics" will announce the winners and answers to Michael Wiesenberg's recent puzzle competition. We will also introduce a new column, "Mac Toolbox," to provide the background needed to write programs that really make use of the Macintosh's capabilities.

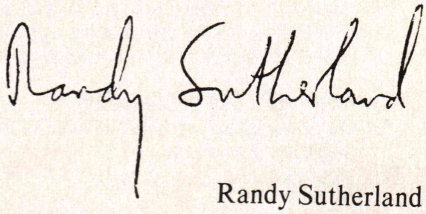
August, we'll C.

In September we'll take a practical look at an academic subject: Algorithms.

October Forth issue: Leo Brodie will report on the implications of the Novix chip for programmers. We also expect to cover Dr. Alan Winfield's Forth machine: an English job, with racing stripes. And we'll review Neon, a Forth-based, Smalltalk-like language for the Mac.

November: Programming Tools for Modula-2 Programmers.

December: Programmer's View of GEM.



Randy Sutherland
Editor

Referees

Ian Ashdown, byHeart Software
Steven Bellovin, AT&T Bell Laboratories

Dr. Dobb's Journal

ARTICLES

Information Age Issues — 52
by Dean Gengle
Advances in telecommunications carry risks to our civil liberties; the problems and some suggested solutions. **(Reader Ballot No. 193).**

Modems: 2400 Bit/Sec and Beyond — 56
by Dale Walsh
Two approaches to increasing modem speed: use more of the available bandwidth; construct more densely packed signaling alphabets. **(Reader Ballot No. 197).**

C UART Controller — 60
by Don Gay
A program for easy modification of the parameters of a programmable USART. **(Reader Ballot No. 195).**

Christensen Protocols in C — 66
by Donald Krantz
This highly-portable file transfer utility is a basic Christensen protocol engine with a minimal terminal program. **(Reader Service No. 196).**

COLUMNS

Dr. Dobb's Clinic — 16
by D. E. Cortesi
The end of shadowy printing; Z80 and BASIC code puzzles; pseudo-random numbers; alloc( ) and free( ). **(Reader Ballot No. 190).**

C Chest — 26
by Allen Holub
Queue management routines; bitmaps for grep.c. **(Reader Ballot No. 191).**

Unix Exchange — 42
by Axel Schreiner
Tricks with the C preprocessor. **(Reader Ballot No. 192).**

16-Bit Software Toolbox — 94
by Ray Duncan
WordStar for the HP LaserJet; Microsoft 8086 MacroAssembler 2.0; the TRIM filter. **(Reader Ballot No. 194).**

CP/M Exchange — 106
by Bob Blum
SID2TTY, an alternate console for SID; CP/M Plus inaccuracy. **(Reader Service No. 198).**

DEPARTMENTS

Editorial 6

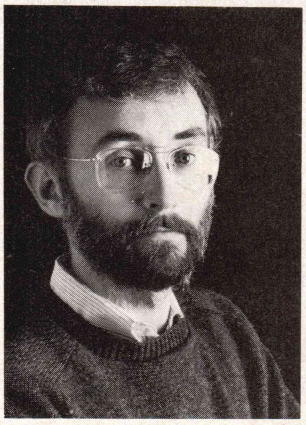
Letters 8

Reviews 92 Artisoft's Envoy

DDJ Classifieds 99

Of Interest 118

Advertiser Index 128



You'll find the answers to this month's Industry Awareness quiz upside down at the bottom of the page.

[1] How many computer books have been published in the last five years with the title *Up and Running*?

[2] How many C compilers are available for the IBM PC?

[3] How many IBM 360/75s were there installed in 1979? (Source: International Data Corporation.)

[4] How many DEC PDP-8s? (Source: ibid.)

[5] How many computer magazines folded in 1984? (Source: Marketing Technology.)

[6] What do you call a computer company that drives away people like Steve Wozniak and Andy Hertzfeld, pretends that the division keeping the company alive doesn't exist and thinks that it can make a machine a home computer or a portable by fiat? (Hint: it starts with an *A*.)

[7] Adam Osborne is predicting that within a year virtually all microcomputer software will sell for under $100 and Phillippe Kahn has been heard making predictions of wealth for Borland personnel that sound suspiciously like those we heard from Osborne Computer Company executives on *60 Minutes* a few years ago. Is Osborne emulating Kahn, or is Kahn emulating Osborne?

[8] By what ratio will Bill "The Cracker" Landreth's royalties for his book *Out of the Inner Circle* exceed the fine he paid when convicted of wire fraud? (Hint: his fine was half the dollar amount of the damage that he says the FBI did to his equipment in confiscating it.)

[9] What do you call an Association for DAta-Processing Service Organizations that responds to the vehement rejection by the Microcomputer Managers' Association of its copy-protection proposal (a plan that MMA threatened to respond to by boycotting participating vendors) by saying "Corporate America would find it hard to turn its back on the [software] these companies are offering"? (Hint: it starts with an *A*.)

[10] Back in February, San Francisco police lieutenant Thomas Suttmeier admitted that he had for the preceding two months had access to all computer files in the city's criminal justice system, including the public defender's files of defense strategies and files of the office that handles civilian complaints against the police. In fact, any police officer with access to the computer could have tapped into the files by typing "Tom's Menus." What can be done to prevent such abuses of power?

Michael Swaine

Michael Swaine

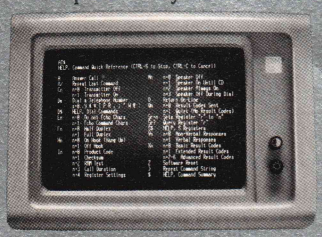
Answers: [1]: At least three. [2]: At least two dozen. We're reviewing them next month. [3] 25. [4] 34,665. [5]: 55. [6]: Arrogant. [7]: Neither. Entrepreneurs do not emulate, imitate, acknowledge predecessors or remember the past. [8]: By conservative estimate, 100:1. [9]: Arrogant. [10]: Be serious. We're talking about the *police* here. They wouldn't do anything wrong.

BIT FOR BIT THE BEST MODEM ON THE MARKET.

Other people make modems for telecommunications. But our new Courier 2400™ modem is made for business. This modern modem transmits, over the phone, 240 characters a second, enabling you to upload or download data at twice the speed of a 1200 bps modem. You'll cut phone costs, save precious hours and increase productivity.

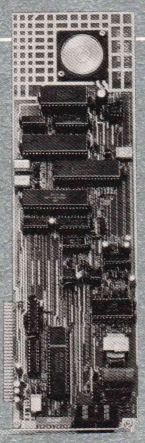
BEING FASTER IS IMPORTANT. BUT BEING BETTER WAS OUR GOAL.

The Courier 2400 features auto-dial and auto-answer ... and is fully CCITT and Bell compatible. It responds to the full AT command set, allowing you to use any of the popular telecom software packages, including Telpac™ by U.S. Robotics, Crosstalk™, PC Talk™, Smartcom™ and many others. And the entire AT command set and S-register functions are displayed on "help screens" and again summarized for you on the underside of the unit.



Help Screens

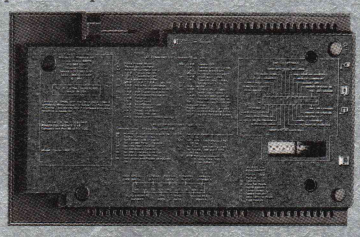
Courier 2400 is accommodating in other ways too. It lets you know the length of each call, tells you (on screen) the status of a call in progress, and even features an adjustable speaker to provide audio phone line monitoring. Courier can test itself in both answer and originate modes, and automatically adjusts from 2400 bps to 1200 or 300 bps. And a powerful automatic equalizer assures nearly perfect performance on every call.



Microlink 2400™

BEST OF ALL, COURIER IS COMPATIBLE WITH YOUR BUDGET.

At $699, you'll not find more modem for the money. If you prefer an internal slot modem for IBM-PC and compatible computers, our new Microlink 2400™ will deliver the same superior performance at the same affordable price. And to get the most out of either Courier or Microlink, ask for new, improved Telpac telecommunications software with easy to use windows.



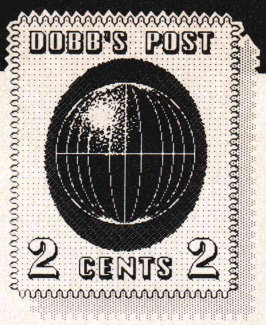
Bottom of Courier

We set out to build the best modem on the market. Now, it's ready. Once you try Courier or Microlink, we think you'll agree — we're not exaggerating one bit.

courier 2400

by U.S. Robotics, Inc.
8100 McCormick Blvd.
Skokie, IL 60076
Phone: (312) 733-0497
Telex: 650-186-3130
Outside Illinois: 1-800-Dial USR

Circle **no. 52** on reader service card.



Prolog

Dear *DDJ*:

I enjoyed reading your March [1985, #101] issue, especially the article entitled "Programming In Logic." I am an avid believer in Prolog, hoping to see it continue to catch on in the U.S.

I'm writing to you in part to thank you for helping promote an understanding of logic programming and Prolog. I am also writing to let you know that while the table of sources of Prolog was a terrific addition to the article, it was not complete. Arity, a spin-off from LOTUS Development Corporation and recently funded by LOTUS, is in the business of building sophisticated Prolog tools and applications.

We take great pride in having built Arity/Prolog, a highly optimized and extended version of Prolog. We have two products available now: the Arity/Prolog Compiler and Interpreter and the Arity/Prolog Interpreter. We are also very excited about our upcoming products.

Sincerely,
Meredith Bartlett
Arity Corporation
336 Baker Avenue
Concord, MA 01742

Dear *DDJ*:

Congratulations on your Prolog issue. The articles, together with Dean Schlobohm's Tax Advisor program, provide an excellent introduction to Prolog.

I would just like to add, for your readers' information, the Quintus name to John Malpas' list of sources of Prolog. Quintus supplies a high-performance Prolog development system for the Sun workstation, and DEC Vax under Unix 4.2 and VMS. Quintus Prolog Release 1.0 features an incremental compiler, integrated Emacs text editor, comprehensive debugger, style checker, C interface, on-line help system, and a high degree of compatibility with DEC-10/20 Prolog and C-Prolog. In addition, Quintus distributes DEC-10/20 Prolog as an unsupported product.

Sincerely,
David Warren
Quintus Computer Systems
2345 Yale Street
Palo Alto, CA 94306

Capitalism vs. Stallmanism?

Dear *DDJ*:

Although I am no fan of Unix, I read Richard Stallman's "Manifesto" with delight. Here is a true adherent to the Hacker ethic! Mr. Stallman's militant advocacy is enhanced by his well-written arguments. Many may find points to dispute in his statement, but all will find it thought-provoking and interesting.

I wish him and his associates luck. The computer industry needs to be turned upside down every so often. Mr. Stallman seems just the fellow to do it this time. Gee—maybe I ought to reconsider my dislike of Unix—I'd hate to be left out of the fun when GNU arrives!

Yours,
James F. Glass
18653 Ventura Boulevard
Suite 351
Tarzana, CA 91356

Dear *DDJ*:

Richard Stallman's "The GNU Manifesto" in your March issue made me furious. I'd love to take 5 pages to dispute all of his 5 pages, but I'll confine myself here to 2 major points.

First, Stallman doesn't believe in property rights. As evidence on a small scale, note the omission of the usual "Unix is a trademark of Bell Labs" in his article, which was present in every advertisement in that issue. Stallman purposely omits this because he doesn't recognize their claim to the name Unix.

However, if it hadn't been for all the time and money invested by Bell Labs in development, then Unix would not be as widespread, popular, or valuable as it is today. If Unix had no value, then neither would GNU. (To see this, reread Stallman's article, substituting "Glorp" for "Unix" and see if he still makes sense.) In effect, Stallman is happy to take full advantage of Bell Labs' investment, with no recompense to them. Would you spend your time and money developing software, only to have Stallman copy it and give it away for free? I wouldn't. (In fact, Stallman wouldn't even give you the choice. He states: "the desire to be rewarded for one's creativity does not justify depriving the world in general of all or part of that creativity.")

Second, Stallman's explicit philosophy is socialist redistributionism. He states that "all sorts of development can be funded with a software tax." Stallman wants to tax *me* to fund *him* to develop software he will give away for *free*. Is this how you want your tax money spent? Not me.

Stallman states: "the fundamental act of friendship among programmers is the sharing of programs"; "good system software [should be] free, just like air"; what deserves a reward is "social contribution." In other words, the Good is what is social, shared, free. On the other hand, he states: "marketing agreements . . . forbid programmers to treat others as friends"; "users [are] at the mercy of one programmer or company that owns the sources"; software vendors

The Ultimate Programmer's Editor

WENDIN'S XTC™

VERSION 3.0 STILL ONLY $99.00

SUPER PROGRAMMERS edit in XTC to make software development a snap! Just look at these powerful features:

MULTITASKING

XTC's built-in multitasking lets you run your macros in the foreground or independently in the background while you continue editing. A background process has full access to editor resources, and can be used to translate code from one language to another in REAL TIME, print files in the background, or even scan syntax while you type in code. Best of all, you can use XTC to edit source and documentation in any programming language!

COMPILE IN WINDOWS

All DOS compilers and utilities can be executed from within XTC using a single keystroke. While it runs, XTC captures your compiler's output and redirects it into your text, so you can compare compiler messages with your source code ON THE SAME SCREEN. And using XTC's macro language, Turbo Pascal is literally only a keystroke away. You can use other compilers and utilities inside XTC too — like Lattice "C," Microsoft Pascal, and IBM's Basic, to name a few.

MACRO LANGUAGE

XTC has the most powerful macro language in the editing world. XTC's macros aren't just keystrokes assigned to keys; they're real programs that can be used to automatically edit source code and data files. Like any real programming language, XTC has control structures like IF THEN ELSE, WHILE DO, REPEAT UNTIL, FOR NEXT, DUPLICATE N TIMES, INDEFINITE LOOP, EXIT, and BREAK LOOP. XTC also has INTEGER, BOOLEAN, and STRING variables to hold numbers, conditions, and pieces of text.

WINDOWS & BUFFERS

With XTC you can display up to 8 different files or parts of the same file on the screen at once. XTC's windows are programmable and can even be linked together to share files. XTC also has 20 other buffers that you can use to hold files and blocks of text.

WORDSTAR COMPATIBILITY

If you already know Wordstar commands, then you don't need to learn a new set of commands. If you want to customize XTC, just write macros to emulate the key layout you're used to. XTC can also read Wordstar files, and can even strip off all of the non-standard high bits with a single command.

LARGE FILE EDITING

XTC lets you edit files entirely in memory (using all available memory), or paged to disk, for maximum flexibility. You can choose how XTC buffers text.

XTC outperforms any other programmable editor on all IBM/PC, /XT, and /AT computers (and true compatibles). If you want to feel the power of XTC before you buy it, just ask for our demo disk (only $10) and try it out. When you buy XTC, we'll knock 10 bucks off the price.

To get your copy of XTC now, order it over the phone — we can ship it the same day you call! Or, send in an order, just like this one:

XTC	**$99.00**
Macro Compiler	**FREE**
Shipping, Handling, Insurance	**3.50**
Want it COD? Add this	**1.90**

TOTAL IT UP, AND SEND IT QUICK!

INCREDIBLE EXTRAS!

- UNDO N TIMES
- REMOVE WORDSTAR HIGH BITS
- EDIT GRAPHICS DISPLAYS
- AUTOINDENTING MODE
- TAB EXPANSION/ COMPRESSION MODE
- EXTRA LONG LINES
- MACRO COMPILER
- TELEPHONE SUPPORT
- 150 PAGES OF DOCUMENTATION
- RUNS ON IBM / PC, XT, AND / AT COMPUTERS (AND TRUE COMPATIBLES)

COMPLETE SOURCE CODE

XTC comes with 7,000 lines of source code jam-packed onto two DSDD disks. Includes 13 modules written in Pascal, and 2 assembly libraries you can use to access the PC's screen, intercept software interrupts (like INT 21H functions), allocate and deallocate memory, and load and execute programs. It's all included FREE for your recreation and enjoyment!

INCLUDED FREE!
Macro Compiler with full source code in "C"! Translates Ada control statements into XTC macros. It even runs in XTC windows!

ORDER HOTLINE
509/235-8088
CREDIT CARDS WELCOME!

WENDIN™

BOX 266
CHENEY, WA 99004

The people who make quality software tools affordable.

Ada® is a registered trademark of the U.S. Department of Defense. Turbo Pascal is a trademark of Borland, Inc. XTC is a trademark of Wendin, Inc.

Circle **no. 112** on reader service card.

"[extract] money from users of a program by restricting their use of it"; "a person who enforces a copyright is harming society." In other words, the Bad is what is individual, owned, bought, sold.

Stallman obviously prefers the former, which is characteristic of the Marxist ideals of the Soviet Union. Me, I prefer the latter, typified by the good old capitalist U. S. of A.

Stallman says he is looking for contributions. I would be delighted to offer a one-way plane ticket to Moscow.

Sincerely,
Robert Schwartz
P.O. Box 1637
Wakefield, MA 01880

Dear Richard:
Hurrah! Three cheers for GNU! Your proposal and the philosophy behind it are so refreshing, I almost couldn't believe what I was reading in this month's issue of *Dr. Dobb's Journal*. As you can see, I am ecstatic that someone is finally doing something about Unix.

Since about 1978 or so I have been going to Unix users' meetings, and I remember meeting Ron Cain at one of these meetings before he left SRI. I obtained a Unix Version 6 system about that time to run on a PDP-11/70 for a company contract. One of my colleagues spent a few nights writing some very useful programs; one was a combination nroff/troff marriage that output bit-mapped fonts on a Versatec printer. Another rotated the nroff/troff output by 90 degrees so that printed copy came out sidewise, but properly formatted on the Versatec 11×8½ fan-fold paper. With that system our little company was in business, and the number of reports and proposals done on that system was amazing. We had to repair the file system now and then, but that was a minor thing compared to the amount of work that got done.

Rather than waste your time with more of my nostalgia, I want to tell you that I have had in mind the same idea as you have proposed. After some of my friends were involved in the secure Unix kernel work at UCLA, I began to realize that it was in fact possible to write a Unix-like system that would not have the royalty problems that make Unix (even on micros) so ridiculously expensive. It is a crime that the micro community does not have access to the hundreds of programs written in C for Unix systems and that so many (inferior) wheels are being reinvented. I obtained a copy of the XINU book and have accumulated a library of articles on other people's experiences in porting Unix to new machines. I had just about everything I needed except a C compiler for my (don't laugh) Commodore 64. Recently one has been announced by Abacus Software, so I am getting all set.

The C64 is a short-term thing; I am looking forward to getting a 68000-based system soon. My motivation for wanting to do a system on the C64 was to be able to run good programs (e.g., a full-screen editor like Emacs) on that system. Memory limitations will obviously rule out multiple users, but if someone comes along with a hard disk interface, at least limited multi-processing would be nice. Already there are switchable memory banks available in 64K sections up to 512K.

For what it's worth, I would like to offer my help in your enterprise. This would most likely take the form of programming assistance. I suspect you will get a lot of other offers of help as well, now that you have "gone public" with your GNU concept. My resources are somewhat limited; however, it may be possible to get a Southern California "GNU Group" going. Perhaps you will spark enough interest to form a mailing list of volunteers.

Once again, I am glad someone like you has come along, and I hope there is some way I can help out.

Sincerely,
Rollin V. Weeks
7130 Marymount Way
Goleta, CA 93117

Dear *DDJ*:
After reading Richard Stallman's "Manifesto" [March 1985, #101], I felt compelled to respond. To begin, I am certainly not opposed to the idea of a public domain version of Unix, nor to public domain software in general. But my initial reaction to Mr. Stallman's sanctimonious outpouring was to feel rather insulted. I have been a programmer for about ten years now, and I don't believe that programmers are any more or any less greedy than others in this society. In the case of software companies supposedly making large sums of money selling copyrighted products, I rather doubt whether the programmers see very much of the profits.

I believe Mr. Stallman must be suffering from a few other fantasies besides that of a public domain Unix. He perhaps forgets that he is living in a capitalist society. Of course, everybody needs food, even more than free software. So food would be distributed for free, right? I have no doubt that farmers, food processing workers, supermarket clerks, truck drivers and others employed in the food industry will not let personal greed blind them to the logic of this argument. People need clothing, housing, medical care and education, too. All these things are necessities and as such should be guaranteed to everyone .... As for Mr. Stallman, I wonder how he can afford to devote so much of his time to this admirable project. Perhaps because his previous employer paid him such a fat salary that he can afford to forget about the dirty business of making a living for a little while.

I am afraid that I might sound almost as self-righteous as Mr. Stallman. I read *Dr. Dobb's* for its valuable technical information and for enjoyment. If any of your readers are concerned about moral or political issues, I would recommend that they subscribe to the *New York Guardian*. And I would also recommend that Mr. Stallman stick to technical matters, for which I am sure he is highly qualified.

Sincerely,
David Kettle
8 Milepost Place, #308
Toronto, Ont. M4H 1E1

Dear *DDJ*:
I read with excitement Richard Stallman's "The GNU Manifesto" until I reached his justifications for his approach. I feel torn because, on the

one hand, I applaud his objective, but, on the other, I cannot accept his rationale of why this is the proper approach.

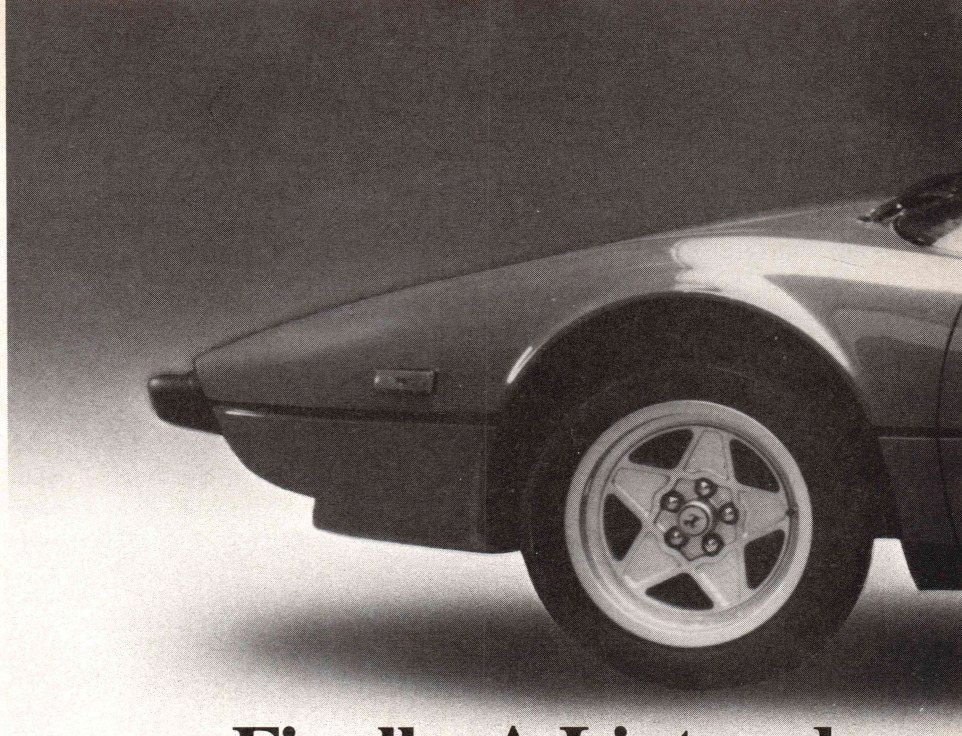
My problem is that he says that "social contribution" should be reward for an individual's creativity. Indeed, he claims that a personal reward is "destructive" because it deprives society of the potential benefits of an individual's creativity. I am a programmer and the owner of a business that sells software and from a selfish point of view would reject his arguments. But I am also a free market advocate and I reject his statements on philosophical grounds which I will attempt to enumerate.

First the wealth of society is increased whenever individuals trade freely. If I charge $100.00 for a program and you freely decide to buy the program, we are both richer. I have my $100.00, which I value higher than the copy of the program, and you have the program, which you value higher than the $100.00. If we were not both richer, then one or both of us would not have entered into the trade.

To say that this is "destructive" to society because, by charging, I deny the benefits of my program to society, is to say that whatever one creates does not belong to him, but belongs in some larger sense to society, and, therefore, to deny it to society is being "destructive."

In a narrow sense Mr. Stallman is right. If Lotus Development gave away "Lotus 1-2-3" instead of charging for it, we would all use it and "society" would be richer. The problem is that Lotus Development would never have created "Lotus 1-2-3" in the first place. Further, the hundreds of others who think they can do it better than Lotus would not go to the effort of attempting to make a better "Lotus 1-2-3" without the promise of financial reward. In this sense society would be poorer because of the free software.

I believe that an individual's physical or intellectual product is his own. Unknowingly, he furthers the general good by selfishly trading it for the most he can get for it. Only a lawyer would say that one's original idea is not his own. Only a dictatorship



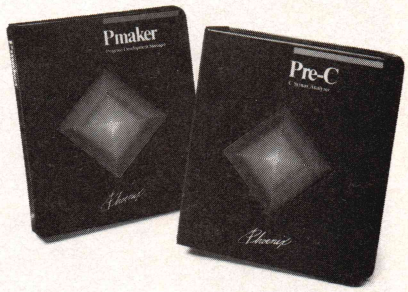
Finally, A Lint and Make for MS™ – DOS

Get the full range of features C programmers working in UNIX™ have come to expect from their Lint and Make utilities. With Pre-C™ you can detect structural errors in C programs five times faster than you can with a debugger. Find usage errors almost impossible to detect with a compiler. Cross-check multiple source files and parameters passed to functions. Uncover interface bugs that are difficult to isolate. All in a single pass. Capabilities no C compiler, with or without program analyzing utilities, can offer. Pre-C outlints Lint, since you can handle analyses incrementally.

Pre-C's flexible library approach lets you maintain continuity across all programs in your shop, whether you use Pre-C's pre-built libraries, functions you already have, or some you might want to buy.

Plus, you're not limited to one particular library. Pre-C keeps track of all the libraries you're using to make sure that code correctly calls them.

With Pmaker™ you can update and track every module in your program. When you make a change in any source or include file, all you do is run Pmaker. It will recompile changed modules and relink your program. With any compiler or linker you choose. Pmaker can update an object module library when one or several of the object modules are changed. You can use Pmaker to handle any task when a change requires several steps.



Pre-C by Phoenix. $395. Pmaker by Phoenix. $195.

Call (1) 800-344-7200.
In Massachusetts (617) 762-5030.

Or, write: Phoenix Computer Products, Corp., 1420 Providence Highway, Suite 115, Norwood, MA 02062.

Phoenix

PROGRAMMERS' PFANTASIES™ BY PHOENIX

Programmers' Pfantasies, Pre-C, and Pmaker are trademarks of Phoenix Computer Products Corporat
MS is a trademark of Microsoft Corporation. UNIX is a trademark of AT&T Bell Laboratories.

Circle **no. 70** on reader service card.

would steal it from the author.

The proof is in the pudding. Today we see hundreds of thousands of programmers working at what they love and doing it to maximize their own rewards. The individual is supreme. The marketplace decides who wins and loses. Society benefits by this enormous effort. If we follow Mr. Stallman's philosophy, we have to elevate "society" to a higher level than the individual. This I cannot do. Indeed, wars have been fought and blood spilled over these very ideas.

Yet, I wish him every success with GNU. I don't know how he keeps food on the table during his efforts, but as he said, his reward is in his love of programming, the pride of seeing his work being used, the recognition of his peers, and just knowing he has contributed something to the world. For some, this is more important than money. Indeed, if given independent means many more would follow the same path.

I hope he succeeds with GNU (I love his recursive name!) because for all the reasons he mentions, I could be a beneficiary. I also wonder whether the success of a "free" operating system will close the door to other operating systems that might be developed if there were some money to be made. He may find that in succeeding in his quest, in the long run he may hurt the "society" that he was hoping to enrich. On second thought, there is nothing to worry about, for if there is a need for a better operating system than GNU in the future, people will be willing to pay for it.

Jim Harlan
Cogitate, Incorporated
24000 Telegraph Rd.
Southfield, MI 48075

Open the door to diskette compatibility with UniForm!

Just $69.95 turns one of your floppy drives into a "look-alike" of most popular CP/M* computers.

UniForm is the easy and inexpensive way to enable your CP/M computer to *directly* read, write and format diskettes from the most popular CP/M micros. This innovative program actually emulates the desired foreign drive. Files that were previously incompatible work flawlessly with your own software tools, such as word processors, database managers and spreadsheets. In addition, you can transfer files to or from MS-DOS.**

Menu-driven UniForm is mastered in just minutes—simply load, select the proper diskette format and go! Operation of your software is unchanged. At just $69.95, compatibility never cost so little!

Uniform is available from your local computer dealer or Micro Solutions.

For users of IBM PC, PC-XT, PC-AT or compatibles, *UniForm-PC* provides direct access to CP/M data diskettes. Also just $69.95.

Trademarks:
*Digital Research
**Microsoft Corporation

MicroSolutions
125 South Fourth St.
DeKalb, IL 60115
815-756-3411

BU Patched

Dear *DDJ*:

I read Jim Rosenberg's Letter to the Editor [March 1985, #101] regarding my article "Archiving Files with CP/M-80 and CP/M-86" [January 1985, #99], and would like to respond to his comments regarding BU and CP/M.

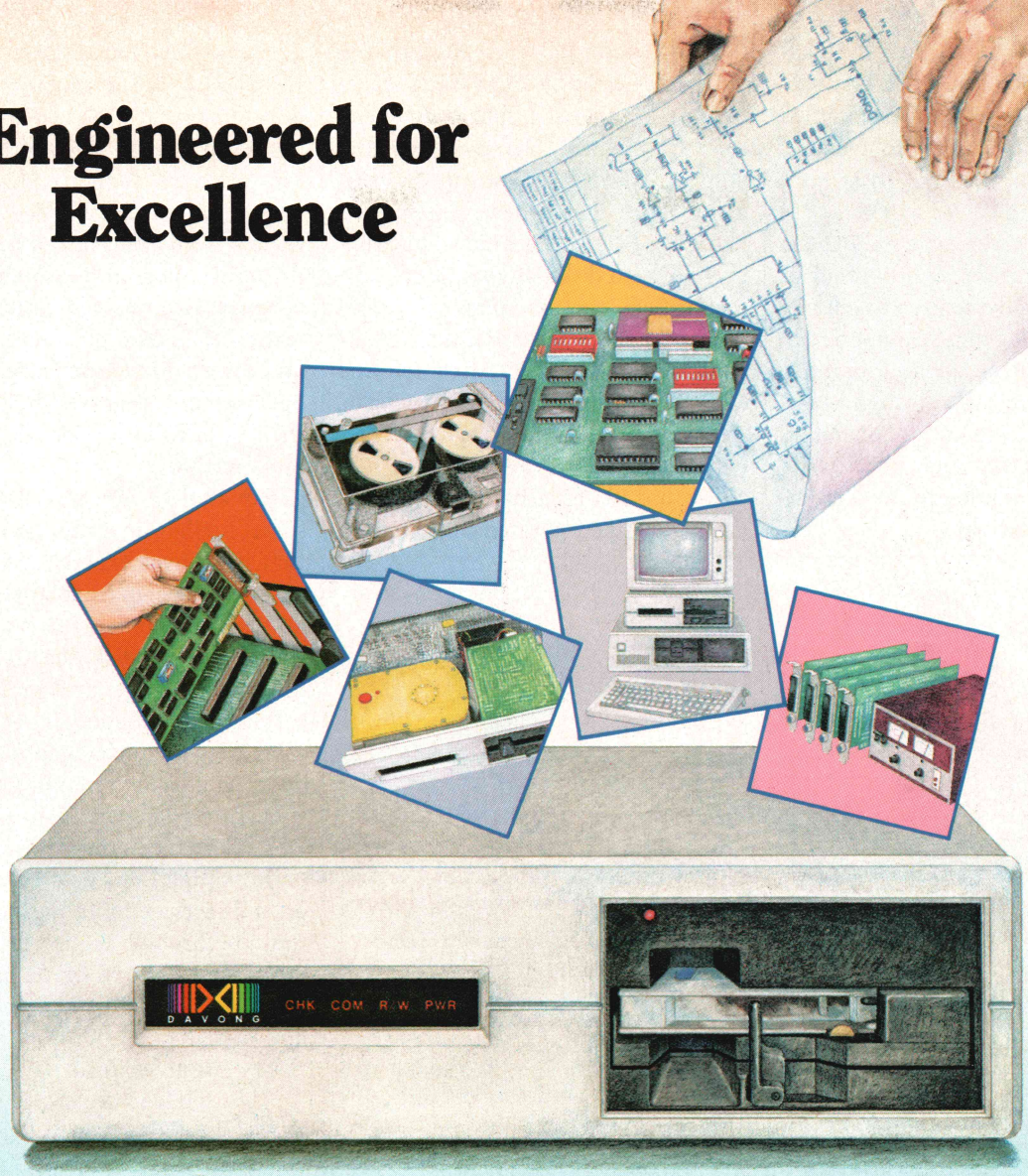
Mr. Rosenberg is correct in stating that the BDOS of CP/M v2.2 does *not* reset file attribute bit t3-prime whenever a file is opened, written to and closed. In this matter I admit to being mistaken. I can also confirm that Dave Cortesi's book *Inside CP/M* is also wrong in stating (p. 222) that "the BDOS sets t3' to 0 whenever it updates a directory entry, that is, whenever the data map of an extent is altered."

However, Mr. Rosenberg chose to ignore the second part of my statement: that a directory entry is changed if it is renamed (or created). In this case the BDOS most definitely sets t3-prime to zero, and it is this fact upon which depends the success of both BU and (presumably) QBAX. This can be demonstrated by setting the file attribute t3-prime of a file with a disk utility, using either the built-in "REN" command or BDOS function 23, "Rename File," to change the file's name, and then examining t3-prime again.

To quote Mr. Rosenberg: "Our own product, QBAX, will function without patching for programs such as word processors and compilers that change files by rewriting them

Circle no. 34 on reader service card.

Engineered for Excellence



We put more into the Davong DataSystem™

so you can get more from your Personal Computer.* You are interested in more productivity. Davong designed the DataSystem for convenience, versatility, and reliability.

Convenience is built-in. Like the package design that slips under your monitor. And installation that's a snap. Put one adapter in a short slot. Connect one cable. Plug it in and go. And you don't need to be an expert to use our software. Menu-driven programs with simple commands and on-line information put you at ease.

Versatility is unsurpassed. DataSystem tape accepts true DOS commands (COPY, etc.) so you can make file-by-file backups. You can even run programs from tape. Or, back up entire disk volumes in minutes with our Tape Manager program. Your data speeds to tape at one million characters a minute. Just data. DataSystem doesn't waste your tape space with non-data areas. And tape capacity is always guaranteed.

Choose from 10, 21, 32 or 43-megabyte disk sizes plus 24 megabytes of removable tape storage. Emulate an XT with Davong Fixed Disk software or enhance your computer's performance with optional Davong Multi-OS® software. And Multi-OS readies the DataSystem for Davong MultiLink™ networking.

Reliability is assured. Only fully-tested components are used in the DataSystem. Before it leaves quality control, each drive must pass days of rigorous system-level tests—and the rack. Each DataSystem must survive a torturous burn-in with no problems. Only then do we consider it good enough for your desk.

If you have the Data, Davong has the System for you. Convenience, versatility and reliability. Why settle for less?

See us at Comdex/Spring, Booth #3232.

*IBM® PC, XT, AT, and many compatibles

IBM® is a registered trademark of International Business Machines Corporation.
Multi-OS®, Davong DataSystem™, and Davong MultiLink™ are trademarks of Davong Systems, Inc.

DAVONG

Davong Systems, Inc.
217 Humboldt Court
Sunnyvale, CA 94089
Phone: (408) 734-4900
Telex: 176386

Circle no. 2 on reader service card.

from scratch." This is exactly what BU will do. Since "BU is almost useless under 2.2 unless BDOS is patched to install support for the archive bit," cannot the same be said for QBAX under a standard CP/M BDOS?

The answer is of course no. BU is a very useful program, as a number of satisfied users have written to tell me. Its one failure is that it cannot detect files that have been updated in place (i.e., changing records through BDOS random access calls or by appending records via BDOS sequential write calls). To do so would require a modified operating system.

QBAX, according to a review by David Fiedler in the October 1983 issue of *Microsystems*, is a fine commercial software product. The review covered QBAX1, which does not support hard disks. Since that review, Amanuensis has released QBAX2, which offers support for hard disks, automatic split file restoration, version numbering and date stamping. Both products apparently include the BDOS patch, referred to by Mr. Rosenberg, that implements the archive bit feature for those files updated in place.

I wrote BU as a basic utility program that incorporated and demonstrated a relatively obscure feature of CP/M v2.2. Readers were invited to add their own enhancements, since space limitations prohibited including features such as QBAX1 and QBAX2 offer. If the *Microsystems* review is anything to go by, I would probably recommend the QBAX products to a business user of CP/M, rather than BU. In this matter, I agree with Mr. Rosenberg. I do not believe that businesses should entrust their data to unsupported public domain utilities.

Regarding CP/M's PIP utility, some further investigation revealed that it behaves exactly as you would expect under the explanation given above. When a new file is created on the destination disk, the archive bit is initially zero. The archive bit can be set by BU or another utility, and thereafter is unchanged by any file operation *except* renaming the file.

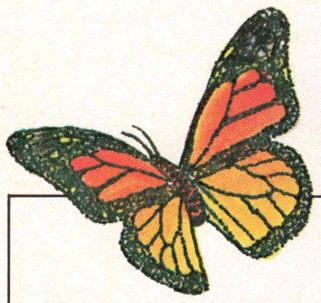
In summary, I wish to thank Mr. Rosenberg for his critique of my article, although I feel he did *DDJ*'s readers a disservice by failing to mention that the archive bit is reset when a file is first created or renamed, and by strongly implying that BU does not work.

Ian Ashdown
byHeart Software
2 - 2016 West First Avenue
Vancouver, B.C. V6J 1G8
Canada

Dear *DDJ*:

We noted with interest Ian Ashdown's BU program published in the January (1985, #99) *DDJ*. We also noted, as Jim Rosenberg of Amanuensis reported in a letter in the March issue, that the "archive bit" t3' is not automatically cleared by the BDOS on file writes or renames.

After examining the BDOS we came up with a patch to accomplish this under CP/M 2.2. With the patch installed, any time a file is written to, either randomly or sequentially, the "archive bit" will be cleared to zero. We have used this with a number of different programs including word processors, SuperCalc and Microsoft BASIC (including writing a single random record) and have encountered no problems with it.



The Language of Artificial Intelligence

PROLOG V

Includes: *interpreter, sample Prolog programs, 122-page manual and primer, and custom-designed binder and slipcase.*

ONLY $69.95

Not Copy Protected

Selected by Japan's Fifth Generation ICOT Project to create machine-based intelligence, Prolog is the most widely used artificial intelligence language worldwide.

POWERFUL AND ELEGANT

With over 70 predefined predicates, PROLOG V complies with the spirit and syntax of Edinburgh Prolog as described in Clocksin and Mellish's classic *Programming in Prolog*, the book that set the de facto standard for Prolog.

IDEAL FOR CREATING:

- ☐ relational data bases
- ☐ expert systems
- ☐ natural language systems
- ☐ biochemical analysis
- ☐ automated design
- ☐ abstract problem solving
- ☐ mathematical logic

THE CHOICE OF UNIVERSITIES

Generous University site licenses have made PROLOG V the choice of Universities from New England to Southern California. Send your inquiry on University stationery, or call for details.

NO RISK OFFER

Examine the package and documentation at our risk for 30 days. Then if you aren't fully satisfied with the quality of the product, return PROLOG V with the distribution diskette still sealed, and we'll refund your money in full.

HARDWARE REQUIREMENTS
- ☐ PC-DOS/MS-DOS 1.4 or later
- ☐ 128 K RAM
- ☐ One disk drive

☐ PAYMENT ENCLOSED $ _____
CA residents add 6% sales tax

☐ CHARGE MY: ☐ MasterCard ☐ Visa

Card No. _____ Exp. Date _____

Signature _____

Mr./Mrs./Ms. _____
(please print full name)

Address _____

City/State/Zip _____

PHONE ORDERS: (619) 483-8513

SHIPPING:
$ 5.00 U.S.
7.50 Canada
10.00 Carribean, Hawaii Air
20.00 Overseas Air

Allow 15 business days for personal and company checks.

COD orders not accepted.

CHALCEDONY SOFTWARE

5580 LA JOLLA BLVD.
SUITE 126 E
LA JOLLA, CA
92037

Circle **no. 21** on reader service card.

Since a number of your readers may be interested in having this facility, I am enclosing a printout of the patch (see Listing, below). This patch assumes that the patch code is located outside of the BDOS, probably in the BIOS. The changes to the BDOS can be made by the BIOS on each warm boot, if necessary, though there are other ways to accomplish this.

Before making the patches to the BDOS, be sure that code to be replaced matches that shown in the enclosed patch (the original code is commented out in the patch).

One other approach which we have used to implement the patch is to install it directly into the BDOS source. We have used C. C. Software's excellent CP/M Source Code Generator to create the BDOS source (we have this software available, as well as our own EUREKA! disk cataloger). We then substituted Z80 instructions in enough places to gather the space we needed for the patch and reassembled the BDOS and patched it directly onto the disk.

We hope that this patch will make BU even more useful to your readers.

Bruce Haanstra, President
Mendocino Software Co.,
P.O. Box 1564
Willits, CA 95490

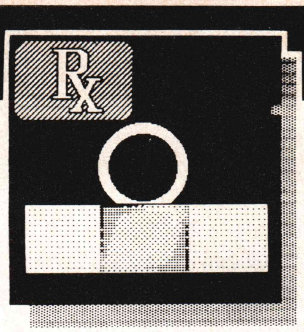
DDJ

Letters Listing (Text begins on page 8)

```
;Desc:Patch to CP/M 2 BDOS to clear update flag on close or rename of a file
;Date:02/02/85
;Time:13:51
;
; The purpose of this patch is to clear the update flag (8th bit of t3) when a
;  file that has been changed is closed.  Part of this patch file will also
;  clear the update flag on a rename.
;
BDOSBASE        EQU     0D800H          ; *** SET THIS TO THE ADDRESS OF THE
                                        ;     BASE OF YOUR BDOS ***

patcharea       equ     00000           ; *** PUT THE ADDRESS WHERE YOUR PATCH
                                        ;     CODE IS LOCATED HERE BEFORE
                                        ;     ASSEMBLY ***

updflg          equ     11              ; offset to update flag byte in fcb
                                        ; 11=t3' (archive bit)

; patch to rename
        org     BDOSBASE+822h
        jmp     patch1
;
; was   lxi     d,16
;       dad     d
;       mov     m,a


; patch to clear update flag on a file that has been changed
        org     BDOSBASE+91Ch
        jmp     patch2
; was   JMP     BDOSBASE+810h   (update1)

;
        org     patcharea
;
; ***** UPDATE FLAG PATCH *****
;  01/06/85 - by BWH
;  this patch forces the directory entry being written to have its update
;   flag bit reset to 0 so FD and MDIR can handle ALL updated files.
;

patch1:                         ; RENAME patch
        lxi     d,16                    ; move over to desired name
        dad     d
        mov     m,a                     ; keep the same user number
        call    clruflg                 ; clear the update flag
        jmp     BDOSBASE+827H           ; (chgnam1) go back and continue

patch2:                         ; UPDATE flag patch
        call    BDOSBASE+55Eh   ; (fcb2hl) get address of fcb in dir buffer
        call    clruflg         ; offset to FD update flag byte (0-11) of fcb
        jmp     BDOSBASE+810h   ; (update1) do original stuff
;
; 01/27/85 - broke out clearing of update flag so rename can do it too
;       IN    -> HL=address of base of FCB with flag to clear
;       USES -> A, HL, DE
;
clruflg:                        ; clear update flag routine
        lxi     d,updflg        ; offset to FD update flag byte (0-11) of fcb
        dad     d
        mov     a,m             ; get the current byte
        ani     7FH             ; clear 8th bit
        mov     m,a             ; and put it back
        ret
;
;
```

End Listing

DR. DOBB'S CLINIC



by D. E. Cortesi

Printer Out of Shadow

In the April column, we asked for help finding competent maintenance for our Diablo printer, but we wrote that column back in February, and the problem wouldn't wait. We posted the same query on several local RBBSs (and very glad we were that the Oxgate, PicoNet, and Simms boards exist). One Eugene Jones came through with a recommendation of The Printer Works (1961 Alpine Way, Hayward, CA 94545; (415) 887-6116). And they did the job: made an informed diagnosis; discussed it with us, explaining exactly what they would do and what it would cost; then did the work. They were thoroughly professional, helpful, and prompt. May they get every broken Diablo in Silicon Valley.

But our problems getting the Diablo fixed have made us wonder: What else is hard to get fixed? Who else is doing a good job fixing things? If you've had maintenance adventures—with happy endings or otherwise—why don't you describe them to us? Especially tell us about repair people who know their stuff. Let us in on the scams, too (with enough horror stories, we can revive "Dobb's Ex Machina"), but it would be super if we could point to good repair services in every major city.

Watt Duzzit Dew

We asked in January David Tilton asked us: What does this short Z80 assembly routine do?

```
BTS:
    cp    (hl)
    ret   z
    rl    L
    djnz  BTS
    ld    L,0
    ret
```

Only Thomas Cage of Crestview, FL, attempted to give us an answer (yea, Tom!), and while he got the essence of it, he was wrong in detail (boo, Tom!). Here is how Tilton described it.

"[The routine] actually outperforms the CPIR instruction of the Z80 in searching a list of bytes for a match by using a binary rather than a linear search, but the restrictions on the sequence of the list and its placement in memory make it less than attractive.

"On entry, A contains the byte to search for, B contains the number of levels in the binary tree, and HL points to the root of the binary tree. One of the unfortunate restrictions is that the value of L for the root must be 01h. Since the value of H remains constant, it is the value of L on return that indicates the result.

"It is helpful to think of the list as a byte array indexed by L. We start with a comparison. If there is a match, the second instruction returns with L indicating where the match occurred.

"If there was no match, the value in the tree was either too large or too small. The carry flag will be set if it was too large. The next instruction multiplies the contents of L by two and, if the carry flag was set, adds one. The DJNZ instruction checks for another level in the tree and, if there is one, loops to check the already-selected node in it. If there is not another level, L is set to zero to indicate that the byte sought was not in the tree.

"There are many disadvantages to this arrangement. It is inflexible. For one thing, a full tree must be maintained; that is, the size of the tree must always be some power of two less one. It is significantly faster than CPIR only for the larger trees. Furthermore, if you can spare a full page of 256 bytes [to hold the tree] then there is an even faster way of doing the same thing:

```
BTL:
    ld    L,A
    ld    L,(HL)
    ret
```

This is, of course, simply a byte look-up table. I thought I was being really clever with my BTS routine, but it turns out to be virtually useless."

Oh, now, cheer up. You got deep into binary trees, right? It's very profound that the successive carry-bits shifted into L are a record of the path through a tree to the desired node, where one means "go left" and zero means "go right." The binary number that defines the path to a node is a unique label for that node; it's basically a "Dewey Decimal Number" for the node.

There's a discussion of this in Knuth's Volume 1, Section 2.3 (especially see the answer to Exercise 15, p. 315). There also is a connection to data compression: if the compared values were wider than a byte, their binary node labels would be a compressed representation of them.

Watt Dozen It Dew

OK, faithful readers, you didn't do too well on Tilton's puzzle; perhaps you can do better with one from David Ross, of Iowa City, IA. He sends the BASIC code you can read in Listing One (page 18). "In theory," he says, "it does something quite remarkable. In practice, it doesn't do much of anything: it is extraordinarily sensitive to roundoff errors. Question: what ought it to do, and to what extent can it be made to work?"

Now, Tilton sent a full explanation

of his puzzle. Ross did not. That means (since the Intern hasn't a clue) that if you don't figure it out and explain it, nobody will ever know the answer.

Random Probing

In February we showed a simple generator of pseudo-random numbers (a PRNG, pronounced "prang"). We're getting mixed feedback on that; at least one reader has written to say it's no good. It would be nice if someone would subject it to some rigorous testing and tell us.

In fact, the letters we are seeing indicate a lot of ignorance about what's good in a PRNG and how to test one. Add to that a paper that Mike Swaine ran across: "Random Number Generation in Microcomputers" by Modianos, Scott, and Cornwell in *Interfaces*, Vol. 14, No. 4 (July-August 1984). We aren't familiar with *Interfaces*; it appears to be a publication of The Institute of Management Sciences.

Anyway, Modianos, et al., studied the characteristics of the PRNGs in various micro BASICs. The net of their findings is "that the random number generators which are intrinsic to the Apple II+ and IIe (using either Applesoft BASIC, Integer BASIC, or CP/M BASIC), the Osborne Executive, and the IBM PC are so flawed that we cannot recommend their use for simulation studies." The only acceptable PRNGs they found were in the HP-86, Apple III, and (oddly) the TRS-80 model III. Plus one for the Apple published by Hare, Faulkner, and Sparks in *Call APPLE*, Vol. 6, No. 1.

These authors apparently expended a lot of time and CPU cycles testing PRNG; unfortunately, they don't discuss their methods in any detail. Sure, that subject has been covered many times before, but not for some years in this magazine. Volunteers?

Object Module

By way of filling up this column, we present a module of C code that has helped us make some quick-and-dirty programs. Sometimes it's easier to do things if you have the ability to allocate a lot of small objects and free them again in any order. In *The C Programming Language*, Kernighan and Ritchie described a pair of functions, alloc( ) and free( ), to do just that. Probably these are in most C libraries.

The compiler we use implements an alloc( ), but contains no free( ). Alloc( ) alone gives the essential ability to allocate chunks of storage for buffers and what-not. But we wanted to make dozens or hundreds of little objects and then discard them again. So we did what you'd do: we rolled our own.

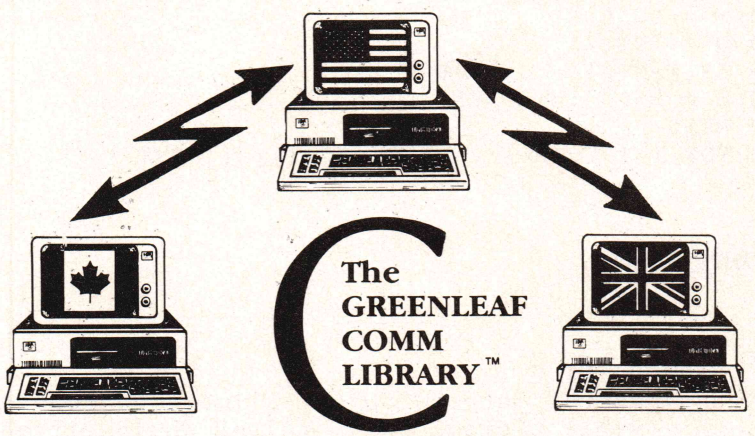
The module for dispensing objects appears in Listing Two (page 18). It duplicates the functions of alloc( ) and free( ), including quite a bit of validation to guard against bugs in

ARE YOU TRYING TO COMMUNICATE ?

C programs can communicate with the world now through the power of **The Greenleaf Comm Library.** Now from the people who brought you **The Greenleaf Functions** General Library for C, comes this rich interrupt driven, ring-buffered asynchronous communications capability.

Over 100 functions in C and assembler to facilitate communications at up to 9600 baud. Up to eight ports at a time. ASCII or XMODEM. X-On/X-Off too. Hayes compatible modems controlled here. Safe too, bet you can't exit your application with interrupts hot. Major applications around the world base their communicating applications on **The Greenleaf Comm Library.** Stop just trying and start really communicating. Get your copy of **The Greenleaf Comm Library** today. For all major C compilers, all models, all versions. For the IBM PC and just about any machine with MSDOS and an 8086. Comes with source code, extensive examples, demo programs, featuring **C-Terminal,** reference card and newsletter. No royalty. $185

Other Products: The Greenleaf Functions General Library, over 220 functions for total control of the IBM PC, with source. $185 for the compilers listed below. (See ordering instructions below).



Specify compiler when ordering: Lattice, Microsoft, Computer Innovations, Mark Williams, or DeSmet. Add $7.00 for UPS Second Day Air (or $5.00 for ground). Texas residents add sales tax. Mastercard, VISA, check, or P.O. In stock, shipped same day.

☐ General Libraries	$185
☐ Comm Library	$185
☐ CI186 Compiler	$349
☐ Lattice C	$395
☐ Mark Williams	$475

PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE.

For Information:
214/446-8641

GREENLEAF SOFTWARE©

2101 HICKORY DR.
CARROLLTON, TX 75006

Circle **no. 43** on reader service card.

the using code. It adds the useful functions of objmax( ) and objtrim( ), which allow you to size an object to match the data it will hold.

The method it uses is not that of Kernighan and Ritchie, but rather an old scheme that you may not have seen before: chaining adjacent blocks by fencing them in with length-words.

The more we use C's #define, the more impressed we are with the flexibility it gives. In the Object module, we used it to define an adjective, WORDAT, to stand for the awkward cast required to say, "the unsigned integer in storage where this character pointer points."

The Object module is definitely written for 8-bit machines; it relies on 16-bit addresses stored in 16-bit integers. The first step in making it portable would be to convert all the hex constants to #defined names. Also, it would be more efficient if it started the search for a free block at the point where it last found or freed one. You might enjoy making those improvements.

The performance of this design won't differ a great deal from that of Kernighan and Ritchie. You might enjoy analyzing the two and working out which is faster, under the assumption of a high turnover of many, small objects.

DDJ

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 190.**

Dr. Dobb's Clinic **(Text begins on page 16)**

Listing One

```
100 REM BASIC PUZZLE BY D. ROSS
105 REM MODIFIED FOR MBASIC BY DEC
110 DEF FNL(X)=LOG(X)/T
120 INPUT "K=",K : DIM A$(K+1)
130 FOR I=1 TO K+1 : A$(I) = "0" : NEXT I
140 T = LOG(10)
150 Z = ASC("0")
160 A0 = K*FNL(2)
170 B = 1
180 WHILE B > 0
190 A = A0+FNL(B)
200 N = INT(A)
210 IF N<0 THEN GOTO 280
220 L = A-N
230 X = 10^L
240 D = INT(X)
250 A$(K-N+1) = CHR$(D + Z)
260 B = B-10^(N-A0+FNL(D))
270 WEND
280 FOR I=1 TO K+1 : PRINT A$(I); : NEXT I
```

End Listing One

Listing Two

```
/*
        OBJECT.C
        general purpose object-allocation module
        (8-bit systems only, relies on 16-bit addresses)

        Copyright (C) 1985 David E. Cortesi
        430 Sherman Ave #212
        Palo Alto, CA 94306
        All rights reserved
        Permission granted to copy for nonprofit purposes.

An "object" is a block of storage of at least two bytes.
Functions in this module manage a pool of storage to
allocate and reclaim objects.

unsigned objinit(u) unsigned u
        Initializes a storage pool of size u, returning the
        size of the largest object that may be allocated.
```

(Continued on page 20)

AZTEC The Most Powerful C

for the IBM AT • MACINTOSH • MS DOS • CP/M-80 • ROM APPLICATIONS
IBM PC/XT • APPLE // • CP/M-86 • TRSDOS • CROSS DEVELOPMENT

Why Professionals Choose Aztec C

AZTEC C compilers generate fast, compact code. AZTEC C is a sophisticated development system with assemblers, debuggers, linkers, editors, utilities and extensive run time libraries. AZTEC C is documented in detail. AZTEC C is the most accurate and portable implementation of C for microcomputers. AZTEC C supports specialized professional needs such as cross development and ROM code development. MANX provides qualified technical support.

AZTEC C86/PRO
— for the IBM AT and PC/XT

AZTEC C86/PRO provides the power, portability, and professional features you need to develop sophisticated software for PC DOS, MS DOS AND CP/M-86 based microsystems. The system also supports the generation of ROM based software for 8088/8086, 80186, and 80286 processors. Options exist to cross develop ROM code for 65xx, 8080, 8085, and Z80 processors. Cross development systems are also available that target most micro computers. Call for information on AZTEC C86/PRO support for XENIX and TOPVIEW.

POWERFUL — AZTEC C86/PRO 3.2 outperforms Lattice 2.1 on the DHRYSTONE benchmark 2 to 1 for speed (17.8 secs vs 37.1) while using 65% less memory (5.8k vs 14k). The AZTEC C86/PRO system also compiles in 10% to 60% less time and supports fast, high volume I/O.

PORTABLE — MANX Software Systems provides **real** portability with a family of compatible AZTEC C software development systems for PC DOS, MS DOS, CP/M-86, Macintosh, CP/M-80, APPLE //+, //e, and //c (NIBBLE - 4 apple rating), TRSDOS (80-MICRO - 5 star rating), and Commodore C64 (the C64 system is only available as a cross compiler - call for details). AZTEC C86/PRO is compatible with UNIX and XENIX.

PROFESSIONAL — For professional features AZTEC C86/PRO is unparalleled.

- Full C Compiler (8088/8086 - 80186 - 80286)
- Macro Assembler for 8088/8086/80186/80286
- Linkage Editor with ROM support and overlays
- Run Time Libraries - object libraries + source DOS 1.x; DOS 2.x; DOS 3.x; screen I/O; Graphics; UNIX I/O; STRING; simulated float; 8087 support; MATH; ROM; CP/M-86
- Selection of 8088/8086, 80186, or 80286 code generation to guarantee best choice for performance and compatibility
- Utility to convert AZTEC object code or libraries to Microsoft format. (Assembly + conversion takes less than half the time as Microsoft's MASM to produce MS object)
- Large memory models and sophisticated memory management
- Support products for graphics, DB, Screen, & ...
- ROMable code + ROM support + separate code and data + INTEL Hex Converter
- Symbolic Debugger & Other Utilities
- Full Screen Editor (like Vi)
- CROSS Compilers are available to APPLE //, Macintosh, CP/M-80, TRSDOS, COMMODORE C64, and ROM based 65xx, and 8080/8085/Z80
- Detailed Documentation

AZTEC C86/PRO-AT $500
(configured for IBM AT - options for 8088/8086)

AZTEC C86/PRO-PC/XT $500
(configured for IBM PC/XT - options for 80186/80286)

AZTEC C86/BAS includes C compiler (small model only), 8086 MACRO assembler, overlay linker, UNIX, MATH, SCREEN, and GRAPHICS libraries, debugger, and editor.

AZTEC C86/BAS $199
AZTEC C86/BAS (CP/M-86) $199
AZTEC C86/BAS (DOS + CP/M-86) $299
UPGRADE to AZTEC C86/PRO $310
C-TREE Database with source $399
C-TREE Database (object) $149

CROSS COMPILERS

Cross Compilers for ROM, MS DOS, PC DOS, or CP/M-86 applications.

VAX -> 8086/80xxx cross $5000
PDP-11 -> 8086/80xxx cross $2000

Cross Compilers with PC DOS or CP/M-86 hosts are $750 for the first target and $500 for each additional target. Targets: 65xx; CP/M-80; C64; 8080/8085/Z80; Macintosh; TRSDOS; 8086/8088/80186/80286; APPLE //.

AZTEC C68K
— for the Macintosh

For power, portability, and professional features AZTEC C68K-c is the finest C software development system available for the Macintosh.

The AZTEC C68K-c system includes a 68000 macro assembler, a linkage editor, a source editor, a mouse based editor, a SHELL development environment, a library of UNIX I/O and utility routines, full access and support of the Macintosh TOOLBOX routines, debugging aides, utilities, make, diff, grep, TTY simulator with upload & download (source supplied), a RAM disk (for 512K Mac), a resource maker, and a **no royalty** license agreement. Programming examples are included. (Over 600 pages of documentation).

AZTEC C68K-c requires a 128K Macintosh, and two disk drives (frugal developers can make do with one drive). AZTEC C68K supports the 512K Macintosh and hard disks.

AZTEC C68K-c (commercial system) $500
AZTEC C68K-p (personal system) $199
AZTEC C68K-p to AZTEC C68K-c upgrade $310
Mac C-tree database $149
Mac C-tree database with source $399
Lisa Kit (Pascal to AZTEC C68k object converter) .. $ 99

AZTEC C65
— for the APPLE //

"...The AZTEC C-system is one of the finest software packages I have seen..." NIBBLE review, July 1984.

The only commercial C development system available that runs native on the APPLE II+, IIc, and IIe, the AZTEC C65 development system includes a full floating point C compiler compatible with UNIX C and other MANX AZTEC C compilers, a 6502 relocating assembler, a linkage editor, a library utility, a SHELL development environment, a full screen editor, UNIX I/O and utility subroutines, simple graphics, and screen functions.

AZTEC C65 (Apple DOS 3.3) $199
AZTEC C65/PRO (Apple DOS + ProDos) $350
(call for availability)

AZTEC C II/PRO
— for CP/M-80

The first member of the AZTEC C family was the CP/M-80 AZTEC C compiler. It is "the standard" compiler for development on CP/M-80. The system includes the AZTEC C II C compiler, an 8080 assembler, a linkage editor, an object librarian, a full library of UNIX I/O and utility routines, CP/M-80 run time routines, the SMALL library (creates modules less than 3K in size), the fast linker for reduced development times, the ROM library, RMAC and M80 support, library source, support for DRI's SID/ZSID symbolic debugger, and more.

AZTEC C II/PRO $349
AZTEC CII/BAS $199
C-TREE Database with source $399
C-TREE Database in AZTEC object form $149

AZTEC C80
— for TRSDOS (Radio Shack Model III & 4)

"I've had a lot of experience with different C compilers, but the Aztec C80 Compiler and Professional Development System is the best I've seen." 80-Micro, December, 1984, John B. Harrell III

This sytem has most of the features of AZTEC C II for CP/M. It is perhaps the best software development system for the Radio Shack Model III and IV.

AZTEC C80 model 3 (no floating point) $149
AZTEC C80 model 4 (full) $199
AZTEC C80/PRO (full for model 3 and 4) $299

To order or for information call:

800-221-0440

(201) 530-7997 (NJ and outside U.S.A.). Or write: MANX SOFTWARE SYSTEMS, P.O. Box 55, Shrewsbury, N.J. 07701.

MANX

TRS 80 RADIO SHACK TRS DOS is a trademark of TANDY.
APPLE DOS MACINTOSH is a trademark of APPLE.

For Technical Support
(Bug Busters) call: 201-530-6557

SHIPPING INFORMATION - Standard U.S. shipment is UPS ground (no fee). In the U.S. one day shipment is $20, two days is $10. Canadian shipment is $10. Two days shipment outside the U.S. is by courier and is freight collect.

Circle **no. 62** on reader service card.

Listing Two

```
        If called a second time, does nothing and returns the
        size of the largest object that can now be allocated.

unsigned objmax()
        Returns the size of the largest object that can be
        allocated from the storage pool as it is now.

char *objget(u) unsigned u
        Allocates an object of size u and returns its address,
        if possible.  If it can't, it aborts the program.

void objfree(p) char *p
        Frees the object addressed by p, which must have been
        allocated by objget().  If it wasn't, or if it is now
        free, or if it's been damaged, the program is aborted.

unsigned objsize(p) char *p
        Returns the size of the object addressed by p.  Aborts
        if *p is not a valid object.

void objtrim(p,u) char *p; unsigned u
        Trims the object addressed by p back to have size u,
        releasing excess space.  Use in a sequence like,
                p = objget(t=objmax());
               ...process data *p to size u, u <= t...
                objtrim(p,u);
        Aborts if *p isn't a valid object or if its size is
        less than u.

void objprt()
        Compiled only if DEBUG is defined, this function
        prints a map of object-space on stderr.

Allocation is first-fit.  Garbage is melded as it is created.
An object is allocated as an even number of bytes L, fenced in
by two words WWWW:

                  A     B               C     D
 ...object (WWWW)(WWWW)<== L bytes ==>(WWWW)(WWWW) object...

Words WWWW are essentially 2+L, where L is the length of the
object they bound.  The low-order bit is the in-use flag.
If (W & 0x0001), the object is active; if not, it is free.
The value K == (WWWW & 0xfffe) == 2+L is used to chain from
block to block.  For addresses A,B,C, and D as shown, A+K->C,
B+K->D, B-2->A, D-2->C, D-K->B, etc etc.  Address B is the
object's address, returned by objget() and given to objfree.

The whole storage pool is bounded by two sentinel words, 0003.
These appear to be active blocks with K=2, hence L=0, and stop
a scan over the pool.

*/
#include "a:stdio.h" /* for error messages */

/*efine DEBUG 1*/ /* or don't -- to exclude objprt() */


 #define WORDAT *(unsigned *)     /* using "char *" ptrs for words */
 #define VOID /* not supported in Aztec C */
 static char *objbase = 0,        /* base of storage pool, uninitialized /
            *objtop;             /* upper limit, for integrity checks */
static unsigned maxobj,          /* largest possible object */
```

```
                j,k;            /* scratch words */
static char *p, *q,             /* scratch pointers */
        minit[] = "Objinit",    /* abort-message parts */
        mget[] = "Objget",
        mtrim[] = "Objtrim",
        mvali[] = "Object validation",
        mpool[] = "Object pool damage";

VOID    objinit(u)
        unsigned u;
{
        if (objbase) return(objmax());

/* ensure an even number of bytes in the entire pool */
        u = (u==0xffff) ? 0xfffe : (u + 1) & 0xfffe;

/* set out 4 bytes for sentinels, 4 for first 2 WWWW words */
        maxobj = u-8;

/* ensure a reasonable size of pool, allocate it */
        if (248 > maxobj)
                objerr(minit,1);
        if (NULL==(objbase = alloc(u))
                objerr(minit,2);

/* set up sentinel words 0003h at ends of pool */
        objtop = objbase + u - 2;
        WORDAT objbase = 0x0003;
        WORDAT objtop = 0x0003;

/* fill pool with a single free object */
        objbase += 2;
        k = maxobj+2;
        WORDAT objbase = k;
        q = objbase + k;
        WORDAT q = k;

        return (maxobj);
}

char *objget(u)
        unsigned u;
{
/* eliminate cases of zero and too-big sizes */
        if (u==0) objerr(mget,1);
        if (u>maxobj) objerr(mget,2);

/* ensure even length, establish enventual K == L+2 */

        k = 2 + (u = (u + 1) & 0xfffe);

/* Scan to the top sentinel, looking for the first free
   object longer or equal to the requested size.  The pool
   is vulnerable to damage --  be very suspicious and test
   every link in the chain before using it. */
        p = objbase;
        while (0x0003 != (j = WORDAT p))
        {
                if (0==(j & 0x0001)) /* free */
                        if (j >= k) break; /* good size */
                q = p + 2 + (j & 0xfffe);
                if ((q <= p)||(q > objtop))
                        objerr(mpool,1);
                p = q;
        }
/* if we hit the sentinel, the requested size is not available
   (could return NULL here instead of aborting) */
        if (j==0x0003) objerr(mget,3);
```

(Continued on next page)

Listing Two

```
/* set p->low WWWW, q->high WWWW, validate block */
        q = p + j;
        if ((q <= p)||(q > objtop))
                objerr(mpool,2);

/* if the wasted space in this block would accomodate
   another object of 4 bytes or more, split off the
   high end as a new, smaller free object */
        if (j > (k+8))
        {
                j = j-k-2; /* j=WWWW of excess object */
                WORDAT q = j;
                q -= j;
                WORDAT q = j;
                q -= 2;
                WORDAT q = k;
                WORDAT p = k;
        }

/* activate chosen block, return address of data */
        WORDAT p += 1;
        WORDAT q += 1;
        return (p+2);
}

unsigned objsize(ob)
        char *ob;
{ /* This is mostly an internal subroutine.  It validates
     a user-supplied object-pointer, leaving static globals
     p, q, and k set up so that p->(WWWW) ... q->(WWWW)
     and k== WWWW & 0xfffe.

        p = ob-2;

/* address must fall within in pool, */
        if (p < objbase) objerr(mvali,1);
        if (p > objtop) objerr(mvali,2);

/* ..must have active lower WWWW word, */
        k = WORDAT p;
        if (0==(k & 0x0001)) objerr(mvali,3);
        k--; /* strip active bit */

/* ..showing size of L=2 or more, but not more than max, */
        if (k<4) objerr(mvali,4);
        if (maxobj < k) objerr(mvali,5);

/* ..which must yield a valid upper-WWWW address, */
        q = p + k;
        if ((q <= p)||(q > objtop)) objerr(mvali,6);

/* ..and the upper and lower WWWWs must match */
        if (WORDAT p != WORDAT q) objerr(mvali,7);

/* object is valid, return its data size */
        return (k-2);
}

VOID    objfree(ob)
        char *ob;
{
/* use objsize() to set up p, q, k and validate block */
        objsize(ob);
```

(Continued on page 24)

YOU CHOOSE:

	MLINK	Crosstalk XVI	Smartcom II
Terminal Emulation Mode			
Menu-driven Interface	Yes	Yes	Yes
Expert/brief Command Mode	Yes	Yes	
Extensive Help Facility	Yes	Yes	Yes
Directory-based Autodialing	Yes		
Automatic Logon	Yes	Yes	Yes
Programmable Function Keys	Yes	Yes	
Multiple Modem Support	Yes	Yes	
File Transfer Mode			
Error Checking Protocol	Yes	Yes	Yes
Wildcard File Transfers	Yes	Yes	Yes
File Transfer Lists	Yes		
XMODEM Protocol Support	Yes	Yes	Yes
Mainframe Versions Available	Yes		
Command Language			
Conditional Instructions	Yes	Yes	
User Variables	Yes		
Labels	Yes	Yes	
Fast Interpreted Object Code	Yes		
Program Run	Yes	Yes	
Subroutines	Yes		
Arithmetic and String Instructions	Yes		
Debugger	Yes		
Miscellaneous			
Remote Access	Yes	Yes	Yes
Electronic Mail	Yes		
Unattended Scheduling	Yes	Yes	
Expandable Interface	Yes		
CP/M, Unix Versions Available	Yes		

MLINK™

The choice is easy. Our MLINK Data Communications System is the most powerful and flexible telecommunications software you can buy for your personal computer. And it's easy to use. MLINK comes complete with all of the features listed above, a clear and comprehensive 275-page manual, and 21 applications scripts which show you how our unique script language satisfies the most demanding requirements.

Choose the best. Choose MLINK.

Corporate Microsystems, Inc.
P.O. Box 277
Etna, NH 03750
(603) 448-5193

Please send me the following version(s) of MLINK:

_____ IBM PC/XT/AT or comparible (DOS, MS-DOS)
_____ DEC Rainbow 100 (MS-DOS)
_____ DEC Rainbow 100 (CP/M-86)
_____ Kaypro 2/4/10 (Please specify) (CP/M)
_____ Wang PC (MS-DOS)
_____ Send me more information

MLINK is available for many systems.
Please call for current version information.

Introductory offer - we pay shipping!
Include $250.00 for each copy of MLINK.
Outside North America add $15.00 (U.S.)

___ Check enclosed ___ MasterCard ___ Visa
___ COD (Registered checks or money orders only)

Card # _____ Exp. _____

Signature _____

Ship to:
Name _____

Address _____

City/State/Zip _____

MLINK is a trademark of Corporate Microsystems, Inc. Crosstalk XVI is a trademark of Microstuf. Smartcom II is a trademark of Hayes Microcomputer Products. IBM is a registered trademark of IBM Corp. MS-DOS is a trademark of Microsoft Corp. CP/M is a registered trademark of Digital Research.

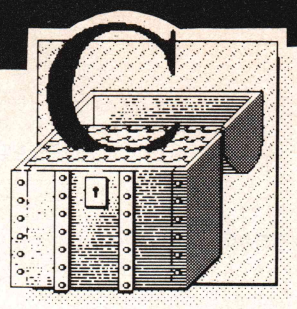
Listing Two

```
/* object is valid, make it free */
        WORDAT p = k;
        WORDAT q = k;

/* if object next above is free, meld with it.  Afterward
   and regardless, ensure that p->KKKK... q->KKKK */
        q += 2;
        j = WORDAT q;
        if (0==(j & 0x0001))
        {
                k = k + j + 2;
                WORDAT p = k;
                q += j;
                WORDAT q = k;
        }
        else q -=2;

/* if block next below is free, meld with it, too */
        p -= 2;
        j = WORDAT p;
        if (0==(j & 0x0001))
        {
                p -= j;
                k = k + j + 2;
                WORDAT p = k;
                WORDAT q = k;
        }
}

VOID    objtrim(ob,sz)
        char *ob;
        unsigned sz;
{
/* validate object, ensure call is to reduce it not enlarge it */
        j = objsize(ob);
        if (sz > j) objerr(mtrim,1);

/* ensure new size will be even */
        sz = (sz + 1)&0xfffe;

/* we will create a new, free object in the high end of
           the present one --  its k is k-sz-4. */
        j = k - sz - 4;

/* if that block doesn't amount to 8 bytes of data, forget it */
        if (j <= 10) return;

/* the k of the existing block will become sz+2 */
        k = sz + 2;

/* create the trimmed block, including active bit */
        WORDAT p = k+1;
        p += k;
        WORDAT p = k+1;

/* create the freed block (no active bit) */
        p += 2;
        WORDAT p = j;
        WORDAT q = j;

/* if its upper neighbor is free, meld them */
        q += 2;
```

```
        k = WORDAT q;
        if (0==(k & 0x0001))
        {
                j = j + k + 2;
                WORDAT p = j;
                q += k;
                WORDAT q = j;
        }
}
unsigned objmax()
{
        p = objbase;
        k = 0;

/* scan the whole pool to find the largest free object. */
/* be suspicious, validate all chain links as we go. */
        while (0x0003 != (j = WORDAT p) )
        {
                if (j & 0x0001)
                        j &= 0xfffe;
                else
                        if (k < j) k = j;
                q = p + j;
                if ((q <= p)||(q > objtop))
                        objerr(mpool,3);
                p = q+2;
        }
        return (k-2);
}

VOID    objerr(m,n)
        char *m;
        int n;
{
        fprintf(stderr,"\n\n%s error #%d\n",m,n);
        exit();
}

#ifdef DEBUG

VOID    objprt()
{
        fputs("\n\nmap of allocation space...\n",stderr);

/* display what should be lower sentinel of 0003, and its address */
        p = objbase-2;
        fprintf(stderr,"%04x->%04x\n",p,WORDAT p);
        p +=2;

/* display bounds and WWWW words of each block to upper sentinel */
        while(0x0003 != (j = WORDAT p))
        {
                q = p + (j & 0xfffe);
                k = WORDAT q;
                fprintf(stderr,
                "%04x->%04x ... %04x->%04x\n",
                p, j, q, k);
                if ((q<=p)||(q > objtop)) break;
                p = q+2;
        }

/* display upper sentinel unless chain was broken */
        if (j == 0x0003)
                fprintf(stderr,"%04x->0003\n",p);
        else
                fputs("...stopped for invalid chain\n",stderr);
}
#endif
```

End Listings

Queues and Bit Maps



by Allen Holub

Queues

I've never been happy with a store-bought word processor. I cut my teeth on vi and nroff on a Unix system, and I've yet to find an acceptable commercial version of either that will run under MSDOS.

Because I had an editor that I could live with, not having nroff was my greatest problem. None of the versions of nroff from users groups were acceptable; they invariably were subsets of the real thing, often loaded with bugs, and at least one of them (roff4) was formatted so poorly that it was unmaintainable. The writers of these programs also took considerable liberties with nroff's instruction set.

So, being a masochist, I wrote my own nroff. In the course of this project, I added several useful functions to my library and these functions will turn up in this column from time to time. This month we'll look at the queue manager .

A queue is a FIFO (first in, first out) data structure. The first object to be enqueued is also the first object to be dequeued (unlike a stack where the first object pushed is the last object popped).

The rate at which objects are enqueued can be higher than the rate at which they are dequeued, as long as you never allow the queue to get full. Therefore, you can use queues to implement type-ahead in character I/O routines and the like. The interrupt service routine that talks to the keyboard UART puts characters into the queue, and getc or its system equivalent takes the characters out of the queue. As long as the queue never fills up, getc doesn't have to keep up with the data rate at the UART.

Queues have other uses. In multitasking operating systems, they hold pending messages while a task is busy. In a word processing application (like nroff), they're used for line filling. Input text to nroff is unformatted: there are a random number of words on a line. Nroff collects words until it has enough to fill an entire line, then it prints the collected line. My version of nroff enqueues characters until it has collected one too many words. It then dequeues all but the last word to produce an output line.

Listing One (page 31) contains a package of queue management routines; you need to declare a pointer to a QUEUE somewhere in your program to use them. Like a FILE, a QUEUE is a structure maintained by the queue manager. However, also like a FILE, your program need know nothing about the details of queue management or the contents of this structure. I usually typedef a QUEUE as a character pointer when I use it. Now let's turn to the various routines.

```
typedef char *QUEUE;

QUEUE *makequeue( qsize, objsize )
int qsize, objsize;
```

Makequeue( ) is the equivalent to fopen( ). "Qsize" is the size of the queue in objects (not bytes), and "objsize" is the size of a single object in bytes. You can make a queue of anything; even a queue of structures is permissible. Like fopen( ), makequeue( ) returns a pointer that must be passed to all the other queue management routines. A zero is returned if there isn't enough memory to make the queue.

```
del_queue( qp )
QUEUE *qp;
int empty;
```

Del_queue( ) deletes a queue created with a previous makequeue( ) call. Qp is the pointer returned by that makequeue( ) call. This routine *only* deletes empty queues. If you need to delete a queue that isn't empty, use free(qp); a one is returned if the queue is deleted, and a zero otherwise.

```
enqueue ( obj, qp )
char *obj;
QUEUE *qp;
```

enqueue( ) puts an object into the queue. "Obj" points at the object to be enqueued. Alhough it's declared as a character pointer, obj actually can point at anything. "Qp" is a pointer to the queue itself, as returned from a previous makequeue( ) call.

```
dequeue ( obj, qp )
char *obj;
QUEUE *qp;
```

Dequeue( ) is the opposite of enqueue( ). It takes an object out of a queue pointed to by "qp" and puts that object at the memory location pointed to by "obj."

```
car *show_next( qp )
QUEUE *qp;

int *sp_used ( qp )
QUEUE *qp;

int sp_avail ( qp )
QUEUE *qp;
```

You use these routines for getting information about a queue. Show_next( ) returns a pointer to the object to be dequeued by the next call to dequeue( ), without actually removing the object from the queue. Sp_used( ) returns the number of objects

now in the queue. Sp_avail( ) returns the amount of space left in the queue. The sum of the return values of sp_used( ) and sp_avail( ) is the queue size.

The QUEUE data structure is a block of memory that includes a header describing the queue and the queue itself. The mechanism is similar to that used by alloc( ) and free( ). This structure is shown in Figure 1 (page 27).

"Start" points at the beginning of the queue proper. "Head" and "tail" are the offsets, in objects, from the start of the queue to the first and last valid objects in the queue. "Size" is the maximum number of objects that the queue can hold. "Nobj" is the number of objects currently in the queue. "Objsize" is the size of an object in bytes.

There are actually more things in the header than are, strictly speaking, necessary. For example, you can determine the start of the queue's data area by adding the header size to the queue pointer. Similarly, you can compute the number of objects in the queue from head and tail. However, these extra fields do reduce the amount of computation you need to do. If RAM is tight and ROM isn't, take them out.

Makequeue( ) (Listing One, line 27) allocates space for the queue and initializes the header. The real queue size is the number of bytes required for the queue itself (qsize × objsize) plus the size of the header.

Del_queue( ) (line 43) just calls free to return the memory used by the queue to the free list (if the queue is empty).

Objects are put into the queue with enqueue( ) (line 57). They are enqueued at the tail and dequeued from the head. The address where the object is to go is computed on line 71. Because qp->start is declared as a character pointer, pointer arithmetic is defeated (if you add 1 to a character pointer, you actually modify it by 1, not so with an integer pointer). The object is transferred into the queue, one byte at a time, on line 75, and the tail pointer is incremented on lines 77-78. If the pointer advances past

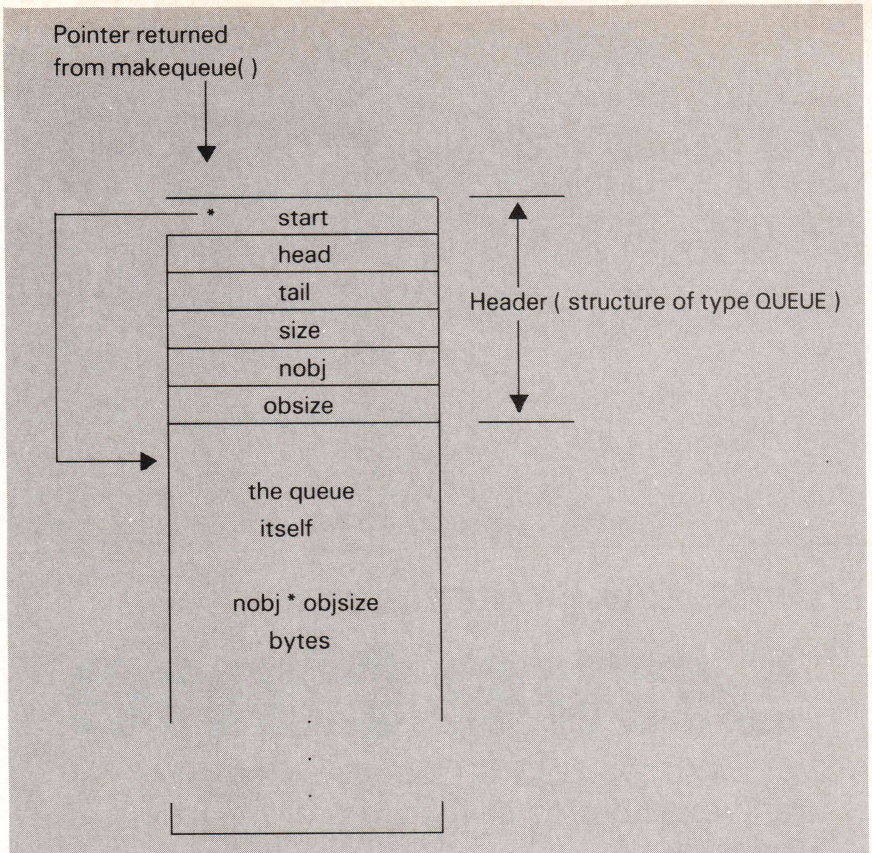


Figure 1
The queue data structure

Does your ISAM run on IBM, APPLE, DEC and AT&T computers? c-tree does, and you only BUY IT ONCE!

The company that introduced micros to B+ Trees in 1979 and created ACCESS MANAGER™ for Digital Research, now redefines the market for high performance, B+ Tree based file handlers. With c-tree™ you get:

- complete C source code written to K&R standards of portability
- high level, multi-key ISAM routines and low level B+ Tree functions
- routines that work with single-user and network systems
- no royalties on application programs

$395 COMPLETE

Specify format:
5¼" PC-DOS 3½" Mac
8" CP/M® 8" RT-11

for VISA, MC or COD orders, call
1-314-445-6833

c-tree™
BY FAIRCOM

2606 Johnson Drive
Columbia MO 65203

Access Manager and CP/M are trademarks of Digital Research, Inc. Apple is a trademark of Apple Computer, Inc. c-tree and the circular disc logo are trademarks of FairCom IBM is a trademark of International Business Machines Corporation DEC is a trademark of Digital Equipment Corporation ©1984 FairCom

Circle no. 37 on reader service card.

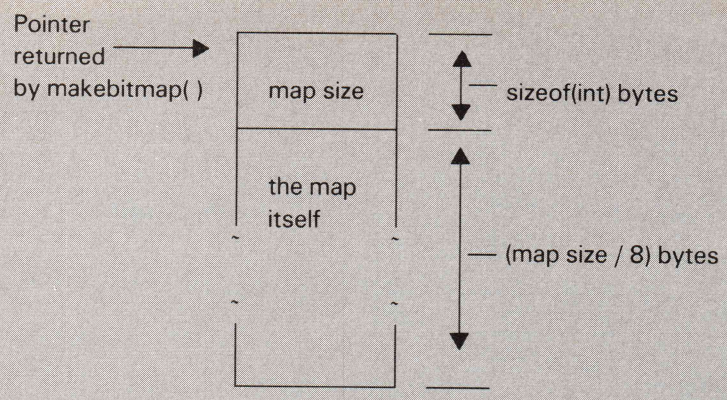


Figure 2
The bit map data structure.

CP/M-80 C Programmers . . .

Save time

. . . with the BDS C Compiler. Compile, link and execute *faster* than you ever thought possible!

If you're a C language programmer whose patience is wearing thin, who wants to spend your valuable time *programming* instead of twiddling your thumbs waiting for slow compilers, who just wants to work *fast,* then it's time you programmed with the BDS C Compiler.

BDS C is designed for CP/M-80 and provides users with quick, clean software development with emphasis on systems programming.

BDS C features include:

- Ultra-fast compilation, linkage and execution that produce directly executable 8080/Z80 CP/M command files.
- A comprehensive debugger that traces program execution and interactively displays both local and external variables by name and proper type.
- Dynamic overlays that allow for run-time segmentation of programs too large to fit into memory.
- A 120-function library written in both C and assembly language with full source code.

Plus . . .

- A thorough, easy-to-read, 181-page user's manual complete with tutorials, hints, error messages and an easy-to-use index — it's the perfect manual for the beginner *and* the seasoned professional.
- An attractive selection of sample programs, including MODEM-compatible telecommunications, CP/M system utilities, games and more.
- A nationwide BDS C User's Group ($10 membership fee — application included with package) that offers a newsletter, BDS C updates and access to public domain C utilities.

Reviewers everywhere have praised BDS C for its elegant operation and optimal use of CP/M resources. Above all, BDS C has been hailed for it's remarkable *speed.*

BYTE Magazine placed BDS C ahead of all other 8080/Z80 C compilers tested for fastest object-code execution with all available speed-up options in use. In addition, BDS C's speed of compilation was almost *twice* as fast as its closet competitor (benchmark for this test was the Sieve of Eratosthenes).

"I recommend both the language and the implementation by BDS very highly."
Tim Pugh, Jr.
in *Infoworld*

"Performance: *Excellent.*
Documentation: *Excellent.*
Ease of Use: *Excellent.*"
InfoWorld
Software Report Card

". . . a superior buy . . ."
Van Court Hare
in *Lifelines/The Software Magazine*

Don't waste another minute on a slow language processor. Order your BDS C Compiler today!

Complete Package (two 8" SSSD disks, 181-page manual): **$150**
Free shipping on prepaid orders inside USA.
VISA/MC, COD's, rush orders accepted. Call for information on other disk formats.

BDS C is designed for use with CP/M-80 operating systems, version 2.2. or higher. It is not currently available for CP/M-86 or MS-DOS.

BDSoftware

BD Software, Inc.
P.O. Box 2368
Cambridge, MA 02238
(617) 576-3828

Circle **no. 12** on reader service card.

the end of the queue, it is reset to the beginning of the queue.

Head and tail are actually offsets (in elements) from the starting address of the queue. In the original versions of these routines, head and tail were pointers (which saved some computation), but the routines were harder to debug and additional computation was added in other places.

Objects are removed from the queue with dequeue( ) on line 82. It works the same way as enqueue( ) except you're now using the head rather than the tail, and you're copying in the other direction (from the queue into the object).

Bit Maps

My version of grep (*DDJ*, October 1984, #96) had several inefficiencies built into it. Some of these inefficiencies were basic to the algorithms used. As several readers pointed out, a state machine implementation would have been faster. However, changing something this fundamental would have meant throwing away most of the program.

Because the thing did work after all, I wasn't tempted to rewrite it completely. Other of the suggested improvements, however, were more reasonable. In particular, Dave Cortesi suggested using a bit map to take care of character classes; his code, which I have transformed into a set of general purpose routines, is presented in Listing Two (page 36). Listing Three (page 37) shows the various changes needed to add bit maps to grep.

Probably the biggest problem with grep.c is that I modeled the program on the version given in *Software Tools in Pascal.* I've never been able to figure out why Kernighan and Plauger did some things the way they did. Pascal does support a pointer type; why don't they ever use it?

Anyway, the problem of character classes is really a problem of set recognition. You want to define a set of characters that are legally in the character class and then to test for membership in that set. The *Software Tools* version creates a string of characters that are legal in a character class, and then searches that

Checks & Balances T.M.

Setting the standard in checkbook and financial management

NEED A FAST, POWERFUL PROGRAM?
Checks & Balances is compiled in assembler, not basic like the other older programs of the Home Accountant vintage.

AT A LOSS BECAUSE YOUR PROGRAM KEEPS ONE MONTH AT A TIME?
Checks & Balances keeps a full year at a time. Print, display or change anything for the year at any time!

FIND AN ERROR IN AN ENTRY FROM 3 MONTHS AGO?
No problem. Put your cursor on it and change it, just like a word processor.

SLOWED DOWN GOING FROM MENU TO MENU TO MENU TO. . . ?
Checks & Balances puts you in charge of your computer. Powerful, easy-to-use English commands let you tell your computer to print, display data, or go to other functions instantly.

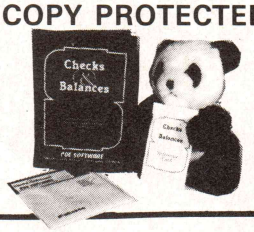
TIRED OF DIGGING THROUGH OLD STUBS TO REMEMBER WHY YOU WROTE THAT CHECK?
Checks & Balances gives you 45 characters of memo space for checks, deposits and cash transactions, and lets you disburse to five accounts. Dollars & Sense makes little sense with its lack of a memo!

WANT A PROGRAM THAT WILL RUN ON YOUR SYSTEM?
Continental told paying customers, "We don't support the Kaypro 4 or 10—too bad!" Dollars & Sense doesn't run under MS-DOS or PC-DOS. But Checks & Balances runs on CP/M, PC-DOS and even the non-IBM MS-DOS machines and APPLE CP/M—Guaranteed! All it needs is 2 floppys or hard disc, 80 × 24 screen and 60K for CP/M, 192K for DOS.

BURNED BY OTHER SOFTWARE?
We have a 30-day money back guarantee and free telephone assistance. No additional warranty fee as with those other programs.

STILL ONLY $74.95—30-DAY MONEY BACK GUARANTEE—NOT COPY PROTECTED

Pick up CHECKS & BALANCES at your local dealer. If he is temporarily out, order directly from CDE. California residents add 6% tax. $3 shipping and handling per order. COD add $4. Outside USA, $10 per order, no COD. Visa and MasterCard accepted. Most orders shipped next day. Include street address for UPS 2nd day air, otherwise, orders shipped by mail.



CDE SOFTWARE • (213) 661-2031
2463 McCready Avenue • Los Angeles, CA 90039

Circle **no. 13** on reader service card.

FOR THE BEST C DEVELOPMENT SYSTEM AVAILABLE FOR THE MACINTOSH, CALL DUVALL.
415-322-2757

Consulair Corp.

Circle **no. 57** on reader service card.

AT LAST

S-100 ⟺ 488

THAT

DOES

EVERYTHING

YOU WANT

IT TO DO

D&W DIGITAL, INC.
20655 Hathaway Avenue
Hayward, California 94541
(415) 887-5711

Circle no. 28 on reader service card.

string sequentially when testing for membership.

Because character classes are often pretty big (i.e., [A–Za–z] has 52 elements), you both waste memory and slow down the program if the character you're looking for is at the end of the list. [A–Za–z] requires 31 comparisons to find a lower-case "e," the most common letter in English.

A better solution to the set problem is a bit map in which each bit corresponds to a single ASCII character; that is, bit 0 is a NULL, bit 1 is a control-A, bit 2 is a control-B, bit 65 is an upper-case A, and so forth. So, to put a letter into the set, you set the corresponding bit in the bit map, and to see if a letter is in the set, you test the corresponding bit.

The set operation will take a little longer than it used to, but you need set a bit only once. The test operation will be much faster, and you generally test bits a lot. Memory use is more efficient too; a bit map that can hold all 128 ASCII characters needs only 16 bytes. An equivalent string would require 129. On the other hand, 16 bytes are always required, even in a small character class.

Listing Two contains three routines:

```
typedef char BITMAP;

BITMAP *makebitmap( size )
unsigned size;
```

This will make a bit map with "size" elements, requiring (size/8 + 2) bytes, and return a pointer to it. Because calloc( ) is used to do the memory allocation, the pointer returned by makebitmap( ) can be passed to free( ) to delete the map. A zero is returned if makebitmap( ) can't get enough memory.

```
setbit ( c, map, val )
unsigned c, val;
BITMAP *map;
```

This will set bit "c" of the bit map pointed to by "map" if "val" is not a zero; it will clear the bit if "val" is zero. C must be in the range 0 to size − 1, where "size" is the argument originally given to makebitmap( ). Map is a pointer returned by a previous makebitmap( ) call. A one is returned on success, and a zero if c is out of range.

```
testbit ( c, map )
unsigned c;
BITMAP *map;
```

This will test bit "c" of the bit map pointed to by "map." It returns a one if the bit is set, and a zero if it isn't or if c is out of range.

Listing Two also contains a main( ) routine for testing the three utility routines (and demonstrating how they work). A bit map containing 32 bits is made on line 66. The contents of the map are printed to the screen with the for loop on lines 74 and 75. The bit is actually set on line 80.

The subroutines are relatively straightforward. Makebitmap( ), like makequeue( ), creates a header with the bit map itself appended to it (see Figure 2, page 28). The pointer returned by makebitmap( ) points at this header, which is a single unsigned sized object (containing the number of bits in the map).

Makebitmap( ) figures the number of bytes needed on line 24 with:

```
numbytes = (size >> 3) + ((size
& 0x07) ? 1 : 0);
```

The right shift does a divide by 8, and the conditional adds 1 to the result of the divide if "size" isn't an even multiple of 8. Calloc( ) is called (on line 28) to get the memory so the bit map will be initialized to all zeroes (all bits cleared). The additional sizeof(unsigned) is for the header. A pointer to the map is returned on line 30.

Setbit( ) starts on line 32 of Listing Two. It tests for "c" out of bounds on line 39 ("map" must be cast into a pointer-to-unsigned to get the map size). We then add sizeof (unsigned) to map to skip past the map size to the map itself. The bit is set on line 43 or cleared on line 45. Testbit( ) (line 49) works just like setbit( ), the only difference being that it returns the bit's value instead of modifying it.

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 191**.

Listing One

```
 1: #define DEBUG 1

 2: #ifdef DEBUG
 3: #include <stdio.h>
 4: #endif

 5: /*-------------------------------------------------------------------
 6:  *      QUEUE.C:        General purpose queue management routines:
 7:  *
 8:  *      Copyright (c) 1985, Allen I. Holub.  All rights reserved
 9:  *  This program may be copied for personal, non-profit, use only.
10:  *-------------------------------------------------------------------
11:  *
12:  *  The QUEUE data structure. No external routine needs to know anything
13:  *  about how this structure is put together. These routines need only
14:  *  remember a pointer to the queue (in a manner similar to the FILE
15:  *  pointer used by the i/o routines).
16:  */

17: typedef struct
18: {
19:         char    *start;         /* Pointer to beginning of queue        */
20:         int     head;           /* Index of current head                */
21:         int     tail;           /* Index of current tail                */
22:         int     size;           /* Max num of objects queue can hold    */
23:         int     nobj;           /* Number of objects now in the queue   */
24:         int     objsize;        /* Size of one element                  */
25: } QUEUE;

26: /*-------------------------------------------------------------------*/

27: QUEUE   *makequeue( qsize, objsize )
28: {
29:         /* Make a queue of the specified size containing objects of the
30:          * specified size. Return a pointer to the queue or 0 if there is
31:          * not enough memory to make the queue. Queues are created using
32:          * calloc(). They require sizeof(QUEUE) + (qsize * objsize) bytes.
33:          */

34:         register QUEUE  *qp;

35:         if( !(qp = (QUEUE *) malloc(sizeof(QUEUE) + (qsize * objsize)) ))
36:                 return 0;

37:         qp->start       = (char *)(qp + 1);
38:         qp->size        = qsize   ;
39:         qp->objsize     = objsize ;

40:         qp->head = qp->tail = qp->nobj = 0;

41:         return( qp );
42: }
43: del_queue( qp )
44: QUEUE   *qp;
45: {
46:         /*   Delete a queue and free the memory. The queue will NOT
47:          *   be deleted unless it is empty. Return 1 if the queue
48:          *   was deleted, 0 otherwise. If you don't care if the queue
49:          *   is actually empty, use free(qp).
50:          */

51:         if( qp->nobj )
52:                 return 0;

53:         free( qp );
54:         return 1;
55: }
56: /*-------------------------------------------------------------------*/
```

(Continued on next page)

Listing One

```
enqueue( obj, qp )
char    *obj ;
QUEUE   *qp  ;
{
        /*      Put an object into the queue. Obj is a pointer to the
         *      object qp is a pointer to a QUEUE. Return 1 on success,
         *      0 if there's no more room in the queue.
         */

        int     i;                                      /* Counter              */
        char    *bp;                                    /* points into queue    */

        if( qp->nobj  >=  qp->size )                    /* If the queue is full */
                return 0;                               /* return failure.      */

        qp->nobj++;                                     /* One more object in   */
                                                        /* the queue            */

        bp= qp->start + (qp->objsize * qp->tail); /* Get target address */
                                                        /* within the queue;    */
                                                        /* then move object     */
                                                        /* into it:             */

        for( i = qp->objsize;  --i >= 0;  *bp++ = *obj++ )
                ;

        if( ++qp->tail >= qp->size)                     /* Wrap around if we've */
                qp->tail = 0;                           /* gone off the end of  */
                                                        /* the queue.           */
        return 1;
}
dequeue( obj, qp )
char    *obj ;
QUEUE   *qp  ;
{
        /*      Get an object from the queue. Qp is a pointer to a QUEUE,
         *      The dequeued object is copied into the place pointed to by
         *      obj. Return 0 if the queue is empty and no object was
         *      dequeued, 1 otherwise.
         */

        register int    i;
        register char   *bp;

        if( qp->nobj <= 0 )
                return 0;                                       /* queue empty */

        qp->nobj-- ;

        bp = qp->start + (qp->objsize * qp->head) ;

        for( i = qp->objsize; --i >= 0; *obj++ = *bp++ )
                ;

        if( ++qp->head >= qp->size )
                qp->head = 0;

        return 1;
}

/*--------------------------------------------------------------------------*/
/*  Little access routines:                                                 */
/*      Show_next returns a pointer to the object at the head of the        */
/*      queue; sp_used returns the number of objects in the queue           */
/*      sp_avail returns the number of slots available in the queue.        */

char    *show_next (qp)
QUEUE   *qp;
{
```

(Continued on page 34)

We're getting hardnosed at Softway.

From now on MATIS™ is only $49.95!

(MATIS, the complete User Interface development tool has been selling for $150.)

Why the radical price cut?

We decided after looking over the competition that MATIS had so many advantages it should be made available to more programmers. We decided to compete aggressively so you could easily afford to have MATIS in **your** bag of tricks. We hear from MATIS users in the USA and France that it is a truly loveable product. Sooo...we're running this big ad to promote our new low price.

MATIS windows are beautiful.

Display any portion of any screens you create at any point in your program. Scroll in any direction manually with cursor keys...or automatically.

And the screens are HUMUNGEOUS!

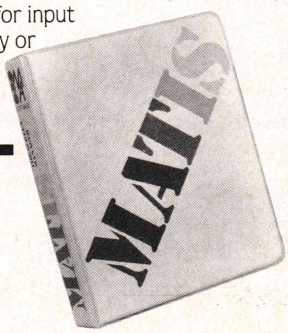
MATIS screens can be just about as big as you want...up to 65,534 rows by 65,534 columns! The number of screens is only limited by available memory.

Print big MATIS screens directly.

One command sends your screens to your printer with no need to program special routines when your virtual screen is bigger than your terminal screen.

User input fields are a snap.

Creating fields for data entry is easy with no limit to size or number by screen. Request for input separately or in blocks.



Denis Moran
President, Softway, Inc.

Control your keyboard with MATIS.

It keeps track of keys that are pressed during the execution of your program and lets you assign specific functions to selected keys.

Control that screen too!

MATIS is extremely versatile and flexible when it comes to controlling lines, columns, fields, and text. They can be modified, transferred, displayed or moved with a single command. All video attributes are supported: color, reverse video, blinking...you name it, you got it.

Want an interactive screen builder?

You've got it with MATIS. It's called "MATPAGE"™ and it lets you create and modify any of your screens in an interactive mode.

MATIS adds over 70 routines to your program.

Written in Assembler, MATIS routines are fast and powerful giving your program improved efficiency and enhanced visual appeal, while they reduce its size and maintenance worries. And MATIS separates screen design from the core of your program.

MATIS is unique.

We don't think there's a single program that combines as many tools in one package as completely or as well as MATIS. It interfaces with Intepreted and Compiled BASIC (Microsoft), C (Lattice, Microsoft, Aztec), PASCAL (IBM, Microsoft) and ASSEMBLER. All you need is an IBM* PC/XT or true compatible under DOS, 128k or RAM, monochrome or color monitor.

You get an easy to follow no-frills manual and a 30-Day Money Back Guarantee.

Late News: MATIS/T™ for TURBO-PASCAL** only $29.95

An indispensable add-on at a dynamite price. What more can we say?

Denis Moran

™MATIS, MATIS/T, & MATPAGE are Trademarks of Softway, Inc.
*IBM is a Reg. Trademark of IBM Corp.
**Turbo Pascal is a Reg. Trademark of Borland International

Softway, Inc.

500 Sutter Street, Suite 222 F
San Francisco, California 94102
(415) 397-4666

24-Hour Credit Card Orders By Phone:
1 (800) 227-2400 EXT 989 In California: 1(800) 772-2666 EXT 989

Please ship the following at once. I understand there is a 30-day money back guarantee.

_____ Copies of MATIS at $49.95 plus $3 shpg. $ _____

_____ Copies of MATIS/T for TURBO PASCAL at $29.95 plus $2 shpg. $ _____

California residents, add 6½% sales tax $ _____

☐ I like to read specs, so send me a folder. Total $ _____

Name _____

Address (please no P.O. Box) _____

City/State/ZIP _____

Phone (_____) _____ Signature _____

Make payment by money order, check, or charge card ☐ VISA ☐ MASTER CARD

Number _____ Exp. _____

Distributed in Europe By: MICRO APPLICATION SOFTWARE • 147 Avenue Paul Doumer, 92500 RUEIL MALMAISON FRANCE, Tel (1) 732.92.54

Circle **no. 92** on reader service card.

 (Listing continued, text begins on page 26)

Listing One

```
112:        return( qp->start + (qp->head * qp->objsize)  );
113: }

114: int    sp_used    (qp)
115: QUEUE  *qp;
116: {
117:        return( qp->nobj );
118: }

119: int    sp_avail   (qp)
120: QUEUE  *qp;
121: {
122:        return( qp->size - qp->nobj );
123: }
124: #ifdef DEBUG

125: main()
126: {
127:        int     num, c, *ip;
128:        QUEUE   *qp;

129:        qp = makequeue( 4, sizeof(int) );

130:        while( 1 )
131:        {
132:                num = c = -1;

133:                ip = (int *)qp->start ;

134:                printf("\n\nqueue: %d %d %d %d\n",ip[0],ip[1],ip[2],ip[3]);
135:
136:                printf("start     =0x%x\n",     qp->start       );
137:                printf("head      =%d\n",       qp->head        );
138:                printf("tail      =%d\n",       qp->tail        );
139:                printf("size      =%d\n",       qp->size        );
140:                printf("objsize   =%d\n",       qp->objsize     );
141:                printf("nobj      =%d\n",       qp->nobj        );

142:                printf("there are %d slots left in the queue\n\n",
143:                                                    sp_avail(qp) );

144:                printf("(d/e/q) ->");
145:                while( c != 'e' && c != 'd' && c != 'q' )
146:                        c = getchar();

147:                if( c == 'e' )
148:                {
149:                        printf("enter decimal number ->");
150:                        scanf("%d", &num );
151:                        printf("enqueue(%d) returned %d\n",
152:                                                num, enqueue(&num,qp) );
153:                }
154:                else if( c == 'd' )
155:                {
156:                        printf( "dequeue returned %d, loaded %d\n",
157:                                                dequeue( &num, qp ), num );
158:                }
159:                else
160:                        break;
161:        }

162:        printf(" deleting queue, queue was %sempty\n",
163:                                        del_queue(qp) ? "" : "not " );
164: }

165: #endif
```

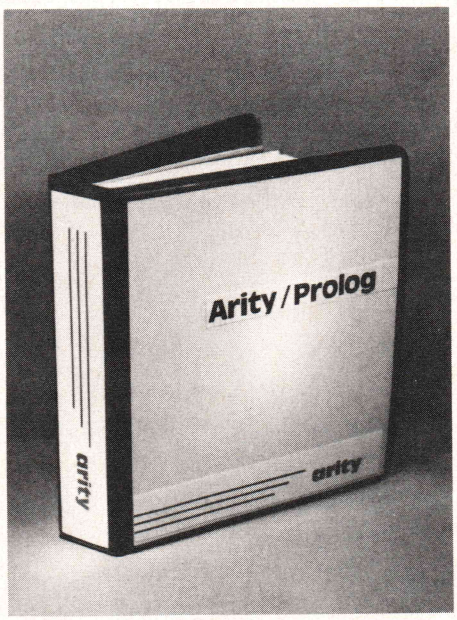
End Listing One

(Listing Two begins on page 36)

More Power Than You Thought Possible

Arity offers the first serious implementation of Prolog for IBM personal computers. Arity/Prolog is a powerful, highly optimized, and extended version of the logic programming language Prolog. Imagine building software applications with a language that solves problems through deduction and logical inference. The task of creating complex programs is much faster and easier, resulting in lower development costs. Arity/Prolog is now in use in a wide range of applications in industry, business, research, and education. The solution—the *Arity/Prolog Interpreter:*

- Source level debugger
- Virtual databases, each with a workspace of 16 megabytes
- Floating-point arithmetic
- String support for efficient text handling



- Interface to assembly language and 'C'
- Text screen manipulation
- Integrated programming shell to MSDOS
- Comprehensive set of evaluable predicates
- Definite clause grammar support

Arity/Prolog Interpreter $495.00

Arity also offers the *Arity/Prolog Compiler and Interpreter,* a sophisticated development environment for building AI applications. Essential for producing fast, serious production code.

Arity/Prolog Compiler and Interpreter $1950.00

The *Arity/Prolog Demo Disk* is available for $19.95. ■ Arity/Prolog products run on the IBM PC, XT, AT, and all IBM compatibles. ■ To order, call (617) 371-2422 or use the order form below.

arity corporation 358 Baker Avenue, Concord, MA 01742

Name _____

Organization _____

Address _____

Quantity	Product	Unit Price	Total Price
	Arity/Prolog Compiler & Interpreter	$1950.00	
	Arity/Prolog Interpreter	$ 495.00	
	Arity/Prolog Demo Disk	$ 19.95	
		Subtotal	
		MA residents add 5% sales tax	
		Total Amount	

☐ Enclosed is a check or money order to Arity Corporation

☐ Please bill my ☐ Mastercard ☐ Visa ☐ American Express

Account # _____

Valid ____/____ to ____/____ signature _____

☐ Please send me more information about Arity and Arity/Prolog

arity 358 Baker Avenue, Concord, MA 01742

M-AD-0

Circle **no. 16** on reader service card.

 (Listing continued, text begins on page 26)

Listing Two

```
 1: /*      BITMAP.C        makebitmap, setbit, testbit: bit map manipulation
 2:  *                      routines.
 3:  *
 4:  *      Copyright (c) Allen I. Holub, all rights reserved. This program
 5:  *      be copied for personal, non-profit use only.
 6:  */

 7: #ifdef DEBUG
 8: #include <stdio.h>
 9: #endif

10: typedef char BITMAP;

11: /*--------------------------------------------------------------------------*/

12: BITMAP    *makebitmap( size )
13: unsigned  size;
14: {
15:         /*      Make a bit map with "size" bits. The first entry in
16:          *      the map is an  unsigned int representing the maximum
17:          *      bit. The map itself is concatenated to this integer.
18:          *      Return a pointer to the map on success, 0 if there's
19:          *      not enough memory.
20:          */
21:
22:         unsigned *map, numbytes;
23:
24:         numbytes = (size >> 3) + ((size & 0x07) ? 1 : 0 ) ;

25: #ifdef DEBUG
26:         printf("Making a %d bit map (%d bytes required)\n", size, numbytes);
27: #endif

28:         if( map = (unsigned *) calloc( numbytes + sizeof(unsigned) ,1 ) )
29:                 *map = size;

30:         return (BITMAP *) map;
31: }
32: setbit( c, map, val )
33: unsigned  c, val;
34: char      *map;
35: {
36:         /*   Set bit c in the map to val.
37:          *   If c > map size, 0 is returned, else 1 is returned.
38:          */

39:         if(  c >= *(unsigned *)map )            /* if c >= map size */
40:                 return 0;

41:         map += sizeof(unsigned);                /* Skip past size   */

42:         if( val )
43:                 map[c >> 3]  |=    1 << (c & 0x07) ;
44:         else
45:                 map[c >> 3]  &=  ~(1 << (c & 0x07)) ;

46:         return( 1 );
47: }


48: /* ------------------------------------------------------------------- */


49: testbit( c, map )
50: unsigned c;
51: char     *map;
52: {
53:         /*   Return 1 if the bit corresponding to c in map is set.
54:          *   0 if it is not.
```

```
55:          */

56:         if( c >= *(unsigned *)map )
57:                 return 0;

58:         map += sizeof(unsigned);

59:         return(  map[ c >> 3 ]  &  (1 << (c & 0x07)) );
60: }
61: #ifdef DEBUG

62: main()
63: {
64:         int     bitnum, set, i, *map;

65:         printf("Making a 32 bit wide bit map\n");

66:         if( !(map = makebitmap( 32 )) )
67:                 printf("Can't make map\n");

68:         while( 1 )
69:         {
70:                 /* Print the bit map. Try to print past the end of the
71:                  * map to make sure overflow detection works (bit 32 should
72:                  * come back as a 0).
73:                  */

74:                 for( i = 0; i <= 32 ; i++ )
75:                         putchar( testbit( i, map ) ? 'X' : '.' );

76:                 printf("\n\nBit number :");
77:                 scanf("%d", &bitnum );
78:                 printf("\nl to set, 0 to clear: ");
79:                 scanf("%d", &set );

80:                 if( ! setbit(bitnum, map, set) )
81:                         printf("Bit out of range\n");
82:         }
83: }

84: #endif
```

End Listing Two

Listing Three

```
 1: /*-------------------------------------------------------------------+
 2:  * GREP:  Changes needed to add bit maps to character classes:       |
 3:  *-------------------------------------------------------------------+
 4:  */

 5: typedef struct token                                 /* In tools.h */
 6: {
 7:         char            tok;
 8:         char            lchar;
 9:         char            *bitmap;
10:         struct token    *next;
11: }TOKEN;

12: /*-------------------------------------------------------------------*/

13: TOKEN   *makepat(arg, delim)                         /* In tools.c   */
14: char    *arg;
15: int     delim;
16: {
17:                                 *
18:                                 *
19:                                 *
20:                 case CCL:

21:                         if (*(arg+1) == NEGATE)
22:                         {
23:                                 ntok->tok = NCCL;
24:                                 arg += 2;
25:                         }
```

(Continued on next page)

Listing Three

```
26:                             else
27:                             {
28:                                     ntok->tok = CCL;
29:                                     arg++;
30:                             }

31:                             if( ntok->bitmap = makebitmap(128) )
32:                                     arg = dodash(CCLEND, arg, ntok->bitmap );
33:                             else
34:                             {
35:                                     fprintf(stderr,"Not enough memory for pat\n");
36:                                     error = 1;
37:                             }
38:                             break;

39:                                     *
40:                                     *
41:                                     *
42: }
43: char    *dodash( delim, src, map )              /* In tools.c   */
44: int     delim;
45: char    *src, *map;
46: {
47:         register int    first, last;
48:         char            *start;

49:         start = src;

50:         while( *src  &&  *src != delim )
51:         {
52:                 if( *src != '-')
53:                         setbit( esc( &src ), map, 1 );

54:                 else if( src == start  ||  *(src+1) == delim )
55:                         setbit( '-', map, 1 );
56:                 else
57:                 {
58:                         src++;

59:                         if( *src < *(src - 2) )
60:                         {
61:                                 first = *src;
62:                                 last  = *(src-2);
63:                         }
64:                         else
65:                         {
66:                                 first = *(src - 2);
67:                                 last  = *src;
68:                         }

69:                         while( ++first <= last )
70:                                 setbit( first, map, 1);

71:                         src++;
72:                 }
73:         }

74:         return( src );
75: }
76: int     omatch (linp, pat, boln)                /* In tools.c   */
77: char    **linp, *boln;
78: TOKEN   *pat;
79: {
80:                                 *
81:                                 *
82:                                 *

83:                 case CCL:
84:                         if( testbit( **linp, pat->bitmap ) )
```

(Continued on page 40)

The first complete programming environment brings the industry to an all-time low.

$80.88

Modula-2 has been hailed as the programming language of the future. Its modular design and built-in error-control features make programming more efficient than ever.

And now there's a system that makes programming more affordable than ever. Interface Technologies' Modula-2 Software Development System (M2SDS).

EASY TO LEARN.
EFFICIENT TO USE.
M2SDS features a "syntax-directed" editor that makes programming easy for beginners to learn. And faster for professionals to use.

With our editor you can enter full statements with a single keystroke. And save up to 90% on typing time. It also gives on-line help in correcting undefined variables and syntax errors—which saves even more time.

Multiple editor windows let you refer to one file while you edit another. That's one more way M2SDS adds hours of more creative, more productive time to your day.

Work faster and easier with multiple editor windows.

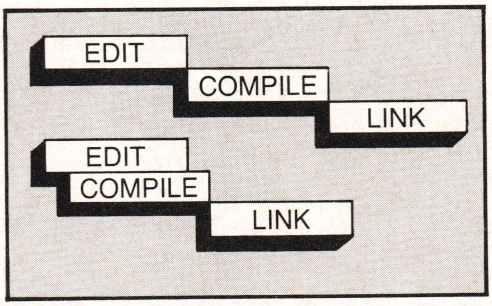
TURN "WAIT TIME" INTO "WORK TIME." When there's no time like real-time, you can count on the M2SDS compiler. Up to 100 lines of Modula-2 text can be turned into native machine code in less than five seconds.

To create programs using your computer's full capacity, there are 18 library modules. And unlike other low-priced compilers, M2SDS has a linker that assembles the components of your program. Automatically.

BREAKTHROUGH TECHNOLOGY.
BREAKTHROUGH PRICE. M2SDS works with IBM® PC, XT, AT or any other 100% compatible computer. Any programs you develop, you own. And M2SDS is non-copy protected.

For just $80.88, M2SDS is the complete programming environment. Including editor, compiler, linker, library modules, 8087 support and more.

Or choose the expanded, fully upgradeable SDS-XP for just $249. Later you can add a debugger, foreign object import module and tool box for even more programming capability. And efficiency.



It not only has a faster compiler, it also saves time by compiling while you edit.

So whether you're a professional looking for a faster way to program, or a novice looking for an easier way to learn, there's a Modula-2 Software Development System just for you.

Call us today for more information or to order your M2SDS. Find out how our new low in system pricing can put your programming efficiency at an all-time high.

WE ACCEPT CHECKS, MASTERCARD, VISA AND AMERICAN EXPRESS. Price does not include shipping and handling. Texas residents add 6.125% Sales Tax. International orders add $30.

INTERFACE TECHNOLOGIES

3336 Richmond, Suite 200, Houston, TX 77098

GET MORE PROGRAMMING EFFICIENCY IN A SYSTEM THAT COSTS LESS. IN TEXAS, CALL (713) 523-8422.

CALL 1-800-922-9049

Circle **no. 50** on reader service card.

 (Listing continued, text begins on page 26)

Listing Three

```
 85:                                    advance = 1;
 86:                            break;

 87:                    case NCCL:
 88:                            if( !testbit( **linp, pat->bitmap) )
 89:                                    advance = 1;
 90:                            break;

 91:                                    *
 92:                                    *
 93:                                    *
 94: }

 95: /*--------------------------------------------------------------------------*/

 96: pr_tok( head )                                        /* In tools.c   */
 97: TOKEN   *head;
 98: {                                  *
 99:                                    *
100:                                    *

101:                    if (head->tok == CCL || head->tok == NCCL)
102:                    {
103:                            printf("string (at 0x%x) =<", head->bitmap );

104:                            for( i = 0; i < 0x7f ; i++)
105:                                    if( testbit(i, head->bitmap) )
106:                                            putchar(i);

107:                            printf(">, ");
108:                    }

109:                                    *
110:                                    *
111:                                    *
112: }

113: unmakepat(head)                                       /* In tools.c   */
114: TOKEN   *head;
115: {
116:            register TOKEN  *old_head;

117:            while (head)
118:            {
119:                    switch (head->tok)
120:                    {
121:                    case CCL:
122:                    case NCCL:
123:                            free(head->bitmap);
124:                            /* no break, fall through to default */

125:                    default:
126:                            old_head = head;
127:                            head = head->next;
128:                            free(old_head);
129:                            break;
130:                    }
131:            }
132: }
```

End Listings

Another in a series of productivity notes on MS-DOS™ software from UniPress.

TEXT EDITING

UNIPRESS EMACS™

Subject: Multi-window full screen editor.

Multiple windows allow several files (or portions of the same file) to be edited simultaneously. Programmable through macros and the built-in compiled MLISP™ extension language.

Features:

- *Famed Gosling Version.*
- *Extensible through the built-in MLISP programming language and macros.*
- *Dozens of source code MLISP functions; including C, Pascal and MLISP syntax checking.*
- *EMACS runs on TI-PC™, IBM-PC AT,™ DEC Rainbow™ or any other MS-DOS machine. Requires at least 384k.*
- *Run Lattice® C or PsMake™ in the background and EMACS will point to any errors for ease of debugging. PsMake is a UNIX™-style "make" utility to automate the process of building complex programs.*
- *Optional Carousel Tools: UNIX-like facilities including cat, cp, cd, logout, ls, mv, pwd, rm, set, sh and more.*

Price:

EMACS	*$475*
One month trial	*75*
Available for UNIX and VMS.	
PsMake	*179*
Carousel Tools	*149*
Full System	*1,299*

Includes EMACS (object), PsMake, Lattice C, PHACT™ ISAM and Carousel Tools.

COMPILER FOR THE 8086™ FAMILY

LATTICE® C COMPILER

Subject: Compiler for MS-DOS.

Lattice C Compiler is regarded as the finest compiler for MS-DOS and is running on thousands of 8086 systems.

Features:

- *Runs on the IBM-PC™ under MS-DOS 1.0, 2.0 or 3.0*
- *Produces highly optimized code.*
- *Small, medium, compact and large address models available.*
- *Standard C library.*
- *PLINK – optional full function linkage editor including overlay and support.*

Price:

Lattice C Compiler	*$425*
PLINK	*425*

ISAM FILE SYSTEM & REPORT WRITER

PHACT™

Subject: Powerful Keyed File Access for MS-DOS.

PHACT ISAM is a keyed B + tree file manager providing easy access to and manipulation of records in a database.

Features:

- *Supports fixed and variable length records (1-9999 bytes).*
- *Up to 9 alternate indices are supported.*
- *Record locking allows each record in the database to allow multiple simultaneous updates.*
- *Records can be accessed on full or partial key.*
- *Includes full Lattice linkable library and high-level functions.*
- *Optional, PHACT-rg, a powerful and flexible report generator which provides a high level, easy-to-use command language for formatting reports from existing PHACT databases. Available for UNIX, MS-DOS and VMS.*

Price:

PHACT ISAM	*$250*
PHACT-rg	*165*

Source Code available, call for terms.

For more information on these and other UNIX software products, call or write: UniPress Software, Inc., 2025 Lincoln Hwy., Edison, NJ 08817. Telephone: (201) 985-8000. Order Desk: (800) 222-0550 (Outside NJ). Telex: 709418. Japanese Distributor: Softec 0480 (85) 6565. European Distributor: Modulator SA (031) 59 22 22.

OEM terms available. Mastercard/Visa accepted.

Trademarks of: Lattice, Lattice, Inc.; UniPress EMACS and MLISP, UniPress Software, Inc.; MS-DOS, Microsoft; UNIX, AT&T Bell Laboratories; Carousel Tools, Carousel MicroTools, Inc.; PHACT, PHACT Associates; 8086, Intel; TI-PC, Texas Instruments; IBM-PC/AT, International Business Machines; DEC Rainbow/VMS, Digital Equipment Corp.

UniPress Software

Your Leading Source for UNIX Software.

by Axel Schreiner

This column is reprinted from the German quarterly unix/mail *(Hanser Verlag, Munich, Germany).* ©1984 *by Axel T. Schreiner, Ulm, West Germany. It may be reproduced as long as the copyright notice is included and reference is made to the original publication.*

/lib/cpp

This column deals with uses and abuses of the C preprocessor. It demonstrates some techniques that can save a lot of work and even more errors. The discussion applies to programming in C in general and makes only very elementary assumptions:

> C programs are run through a preprocessor *before* they are handed to the actual compiler. The preprocessor performs (parameterized) text substitution (#define), inserts header files (#include), and can exclude parts of the source from compilation (#if).

Because the preprocessor is independent of the actual compiler—and does not know C at all—one can use it in particular to extend the C language. Only taste limits one's imagination here . . . .

Excluding Text

Every programmer writes occasional comments. Sometimes we comment to exclude program parts from a compilation. Because in Standard C comments may not be nested, there is considerable temptation not to comment such excluded program parts anymore.

The following technique for text exclusion is much more appropriate:

```
#ifdef not_defined
   crash_the_system(NOW);
   /* this definitely goes wrong */
#endif not_defined
```

Of course, the name not_defined should really not be defined.

Vector Dimensions

In principle, one can determine the size of a vector by using the sizeof operator. However, sizeof yields the size in bytes, not in elements. The following macro determines the number of elements in an arbitrary vector:

```
#define DIM(x) (sizeof (x) / sizeof ((x)[0]))
```

sizeof does not really need parentheses, if it is used to determine the size of an object and not of a data type. One should, however, enclose macro parameters in parentheses. Then things work out for a vector with more than one dimension, too:

```
main( )
{
      struct { int a; char b; } v[10] [20] [30];
      printf("%d %d %d\n", DIM(v), DIM(v[1]),
            DIM(v[1] [2]));
}
```

The program produces the values 10, 20, and 30.

Parentheses should not be necessary in this use of sizeof because a vector subscript should have precedence over sizeof. However, my copy of the Mark Williams CP/M-86 C compiler does not seem to know this.

We can carry these ideas somewhat further. The last element of a vector is

```
#define LAST(x) ((x)[DIM(x)-1])
```

and the customary for loop is, for example:

```
#define END(x) ((x) + DIM(x)-1)

int vector[10], * vp;
   . . .
   for (vp = vector; vp <= END(vector); ++ vp)
      . . .
```

The compiler evaluates sizeof during its evaluation of constant expressions. This circumstance can be used to determine the length of constant strings in an efficient and flexible fashion:

```
#define STRLEN(s) (sizeof (s) - 1)

char buf[STRLEN("model") + 1];
   . . .
   strcpy(buf, "model");
```

There is the danger, however, that STRLEN is used for other objects (i.e., non-strings) by mistaking it for strlen.

Trace

It is well known that a *macro call* is not recognized in a

constant string. Less well known, but more useful perhaps, is that a *macro parameter* is recognized and replaced within the replacement text of a macro definition. Rather than

```
printf("variable = %d\n", variable);
printf("formula = %f\n", formula);
```

we write

```
#define SHOW(val,fmt) fprintf(stderr,\
        "SHOW: val = fmt\n", val)

        SHOW(variable, %d);
        SHOW(formula, %f);
```

The latter is easier to use and conveys more information because val is replaced in the format by the entire macro argument.

A bit of caution is required: if the % operator is used within val, there will be problems with the format. This can be corrected as follows:

```
#define SHOW(val, fmt)  fprintf(stderr, "%s\
   = fmt\n", "val", val)
```

A macro can be defined without a replacement text. Uses of SHOW thus can be eliminated easily from the compiled program altogether. Alternatively, we can specify a condition:

```
#ifdef   DEBUG
         char debugflag;
#        define SHOW(val,fmt) (debugflag &&\
               fprintf( . . . ))
#else    ! DEBUG
#        define SHOW(val,fmt) /* null */
#endif   DEBUG
```

In this example, SHOW is always used as a statement and not as an expression. Using && rather than if has two advantages: we do not *have* to use SHOW as a statement, and use of SHOW does not invite an unintentional else. debugflag, by the way, should be used as a bit vector, e.g.:

```
#define SHOW(level, val, fmt) (debugflag &\
   1<<level && fprintf( . . . ))
```

Now we can maintain different sets of trace information at levels 0 through 7.

Global Variables

Do you like modular programs with lots of sources, makefile, a central header file, and the (feeble) hope that all global declarations really match? Do you like to lint, too? The following technique simplifies maintaining global variables. The central header file contains about the following:

Only $95 with FULL SOURCE CODE!

Q/C

"... an incredible learning tool." *Byte*

For only $95, Q/C is a ready-to-use C compiler for CP/M with complete source code. Here's what *BYTE* (May 1984) said: "Q/C ... has a portable library and produces good code quality. If you want to learn compiler construction techniques or modify the standard language, Q/C is the obvious choice."

- Source code for compiler and over 75 library functions.
- Strong support for assembly language and ROMs.
- No license fees for object code.
- Z80 version takes advantage of Z80 instructions.
- Q/C is standard. Good portability to UNIX.

Q/C has casts, typedef, sizeof, structure initialization, and function typing. It is compatible with UNIX Version 7 C, but doesn't support long integers, float, parameterized #defines, or bit fields. Call about our new products: Q/C profiler, Z80 code optimizer, and Z80 assembler and virtual linker, all with full source code!

THE CODE WORKS

5266 Hollister, Suite 224
Santa Barbara, CA 93111
(805) 683-1585

Q/C, CP/M, Z80, and UNIX are trademarks of Quality Computer Systems, Digital Research, Zilog, Inc., and Bell Laboratories respectively.

Circle **no. 23** on reader service card.

Circle **no. 84** on reader service card.

```
#ifndef   GLOBAL
#         define GLOBAL extern
#endif    GLOBAL

GLOBAL  int   global_variable;
```

If GLOBAL is not defined, a variable declared GLOBAL is declared extern.

Within exactly *one* of the source files that include the header file, we have to take care that the variables that were declared extern elsewhere are really defined. In the main source file, we therefore write:

```
#define GLOBAL /* to define global variables */
#include "definitions.h"
```

One can even initialize global variables in this context *without* resorting to the −m flag instructing the loader ld to accept multiple definitions:

```
#ifdef    GLOBAL
#         define    INIT(x) = x
#else     ! GLOBAL
#         define    GLOBAL extern
#         define    INIT(x)
#endif    GLOBAL

GLOBAL  int variable INIT(10);
```

This technique is not very practical for aggregates. The following variant is easier to use:

```
#ifdef    GLOBAL
#         define    INIT(x) =x
#         define    GINIT
#else     ! GLOBAL
#         define    GLOBAL extern
#         define    INIT(x)   ;
#         undef     GINIT
#endif    GLOBAL

GLOBAL  struct { int a; char b; } variable INIT( )
#ifdef    GINIT
                    { 10, 'b' };
#endif    GINIT
```

This method requires that the C preprocessor permit a macro call with an empty argument list and that the C compiler not complain about superfluous semicolons between global declarations. This method is admittedly no longer very elegant, but it has the significant advantage that the text of central definitions exists only once in all cases.

/bin/lex

Now you see it . . .

lex programs have lots in common with fashions: the effect is not always what the pattern promises. If a function generated by lex is used as a front end for a parser generated by yacc, it is sometimes very hard to decide where to place the blame for a bug: is there a bug in the grammar presented to yacc, or are the patterns that were processed by lex at fault?

The following technique[1] permits the construction of a source file for lex that is conditionalized so that a debugging version can be compiled at any time without any changes to the source. To test the results of lex, all inputs that the parser is to receive later are first presented to the debugging version. This version of the front end then prints a mnemonic version of the values that the parser would receive:

```
%{
#ifdef  TRACE

#       include "assert.h"

        main( )
        {
                char *cp;
                assert(sizeof(int) >=
                   sizeof(char *));
                while (cp = (char *) yylex( ))
                   printf("%-.10s is \"%s\"\n",
                      cp, yytext);
        }

#       define token(x)  (int) "x"

#else   ! TRACE

#       include "y.tab.h"
#       define token(x)  x

#endif  TRACE
%}
```

Normally, TRACE is undefined, and the tokens (i.e., the values that are to be returned to the parser) are defined in the file y.tab.h generated by yacc as:

```
#define NAME     257
        . . .
```

These defined names are used directly in the source presented to lex and are returned as a result of the function yylex( ).

If TRACE is defined, y.tab.h need not yet exist. In this case (i.e., in the debugging version), we want to return a string as a result of yylex( ) that is then printed by the main( ) program included here.[2] Analyzing the debugging output is most easily accomplished if the output uses exactly those words that later will appear in y.tab.h—i.e., that are a result of %token statements in the source presented to yacc.

We are using the fact that macro parameters are replaced within strings in the replacement text of a macro. token(x) returns either x itself (to be passed on to yacc) or

CP/M SYSTEM INVENTORY LIQUIDATION

Old operating systems never die, they get better with age! Multitech Electronics, a large manufacturer of complete computer systems has an inventory of CP/M computers that must go. These fully guaranteed high performance units are new factory packaged systems with the CP/M 2.2 operating system, 64K of RAM, dual floppy disk drives, and CBASIC programming language. You'll get more than what you paid for with complete systems at bargain prices.

FULL CP/M SYSTEM SPECS

	MIC-501	MIC-504
CPU	Z-80A	Z-80A
RAM MEMORY	64K	64K
STORAGE	500K BYTES	2M BYTES
SERIAL COMMUNICATIONS	DUAL RS—232C PORTS	DUAL RS-232C PORTS
PARALLEL COMMUNICATIONS	SINGLE PARALLEL PORT	SINGLE PARALLET PORT
PRICE	$500	$700

This low price also includes: full compatibility with the wide bank of CP/M software, CP/M 2.2 operating system with utilities and manuals, CBASIC 2 programming language, as well as a User's Manual and Service Manual. In addition you'll get a 30 day full warranty. And Multitech will also throw in a Liberty Freedom 100 terminal for $400.00 with the purchase of an MIC-500 system. Call Multitech today for more on your CP/M solutions. Cash or credit cards only. Limited quantities available.

Multitech ELECTRONICS INC.

195 WEST EL CAMINO REAL
SUNNYVALE, CALIFORNIA 94087
TELEX: 755042 MAC SUVL
IN CALIFORNIA, CALL: (408) 773-8400
OUTSIDE CALIFORNIA, CALL: (800) 538-1542

Circle **no. 42** on reader service card.

a string "x" for the purposes of TRACE.

The remainder of the lex program is now quite obvious:

```
%%

[0-9]+                  return token(NUMBER);
[a-z_A-Z][a-z_A-Z0-9]*  return word( );
[ \t\n]+                ;
.                       return token(yytext[0]);

%%

struct reserved { char * text; int yylex; } reserved[ ] = {
    { "begin", token(BEGIN) },
    { "end", token(END) },
    (char *) 0 };
int word( )
{

    struct reserved *rp;
    for (rp = reserved; rp->text; ++ rp)
        if (strcmp(yytext, rp->text) == 0)
            return rp->yylex;
    return token (NAME);
}
```

Yes, there should have been a binary chopped search, but we are dealing only with the principles.

/usr/src/main.c

Argument Standards

Command arguments are always good for surprises. Sometimes several options may be combined into one argument; sometimes each option must be a separate argument; sometimes a parameter value follows as part of the argument; sometimes it does not; sometimes all of the above; sometimes some of the above . . . .

If one consults the sources of certain Unix utilities, one learns to appreciate the flexibility of C (or the infinite patience of the C compiler). Everybody does his or her own thing, and most do it differently in every program! However, it would be so simple to develop a standard as in Listing One, page 48).

At show( ), argc contains the number of arguments that have not yet been processed, and *argv is the first one of these. This argument can be a single – character; in some ancient (cat) and almost new (tar) utilities, this indicates that standard input or output is to be used in place of a file argument.

Flags can be combined at will. If an option requires a value, it can follow immediately or it can be an argument of its own.

Following a standard proposed in the "USENIX login," an option – – serves to terminate processing of the option list. Apart from that, options must start with – and must precede other arguments. These rules, however, still do not cover all possibilities of pr.

The skeleton above is useful but anatomically somewhat terrifying. The following incarnation is perhaps more attractive:

```
#include <stdio.h>
#include "main.h"

#define show(x) printf("x = %d\n", x)
#define USAGE fputs("cmd [-f] [-v   #]\n",\
    stderr ), exit(1)

MAIN
{
        int f = 0, v = 0;

        OPT
                ARG 'f':
                        ++ f;
                ARG 'v':
                        PARM
                        v = atoi(*argv);
                        NEXTOPT
                OTHER
                        USAGE;
        ENDOPT
    show(f), show(v), show(argc);
    if (argc)
        puts(*argv);
}
```

The trick, of course, is concealed in the header file main.h: here the macros OPT, ARG, PARM, NEXTOPT, OTHER, and ENDOPT must be defined using exactly those texts that were given explicitly in the previous example, as in Listing Two, (page 50).

The definitions are not exactly beautiful, especially if they need to be compacted so that the C preprocessor accepts the lengthy replacement texts, but they need to be developed only once to make the argument standard available for all applications. An application then is almost self-documenting:

MAIN
is the function header of the main program.

OPT
starts the loop during which the options are processed.

ENDOPT
completes this loop.

ARG
within the loop starts the processing of one option; the name of the option (a single character) enclosed in single quotes and a colon must follow.

PARM
follows the option specification if the option has a value parameter; the parameter itself is then available as *argv.

BRIEF™

"BRIEF Is The Best Editor I Have Ever Used. I Switched. I Use BRIEF Regularly."

Steve McMahon*

Choosing the right program editor is an extremely important decision. After all, more time is spent with your program editor than with any other category of software. A good program editor helps to stimulate creativity and produce programs of superior design. A poor program editor, on the other hand, can be worse than none at all. It can slow you down, get in your way, and make even the simplest tasks difficult.

BRIEF – PROGRAM EDITING *YOUR WAY*

Every programmer has an individual style that makes their work unique. Most program editors compromise that style by forcing the programmer to conform to their methods. Not so with BRIEF.

BRIEF's most powerful feature is its ability to conform to your way of programming. BRIEF can be easily tailored/customized to your individual preferences. For example, keyboard reassignment allows the keyboard to be used in whatever way you prefer. Keystroke macros allow long sequences of editor directives to be repeated by pressing a single key. Plus you can simultaneously work with numerous program and data files, and configure many windows to control BRIEF's visual presentation.

75% OF THE EXPERTS WHO HAVE TRIED BRIEF SWITCHED! (Call For Details)

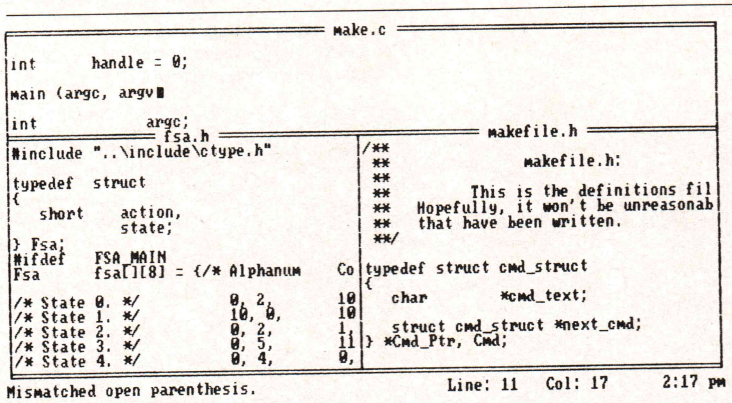
"A BONA FIDE UNDO" Steve McMahon – BYTE REVIEW "BUILD YOUR DREAM EDITOR" MARCH 1985

Here are Steve McMahon's exact words: "...BRIEF implements a true undo facility, by default allowing command-by-command recovery from the last 30 commands...The number of commands you want to be able to undo can be changed." (up to 300) "Only with BRIEF, though, was it possible to undo a macro that produced 4000 words of text with a single keystroke."

EVERY FEATURE YOU WANT

BRIEF supports practically every feature you've ever seen, and some you probably haven't thought of yet.

- Edit Multiple Large Files
- Windows (Tiled & Pop-up)
- Unlimited File Size
- Full UNDO (N Times)
- True Automatic Indent for C
- Exit to DOS Inside BRIEF
- Compile programs inside BRIEF
- Uses All Available Memory
- Intuitive Commands
- Tutorial
- Repeat Keystroke Sequences
- 15 Minute Learning
- Reconfigurable Keyboard
- Online Help
- Search for "Regular Expressions"
- Mnemonic Key Assignments
- Horizontal Scrolling
- Comprehensive Error Recovery
- And ... a Complete, Compiled, Programmable and Readable Macro Language.



A TYPICAL BRIEF SCREEN

Notice there are three windows on the screen simultaneously and each one is viewing a different file. The mainline of a C program is visible in the uppermost window; the programmer has run a syntax checking macro which found a mismatched open parenthesis in the arguments to the mainline. The other two windows show header files containing information crucial to the design of the program. BRIEF can have an unlimited number of windows and files accessed simultaneously.

BRIEF'S ONLY LIMITATIONS ARE THE ONES *YOU* SET

BRIEF's capabilities are limited only by your imagination. Once you've used BRIEF, you will NEVER want another editor.

TRY BRIEF RISK FREE FOR 30 DAYS!

Order BRIEF ($195) and try it for 30 days with a full money-back guarantee. CALL 800-821-2492

AVAILABILITY: IBM-PC and Compatibles, AT, & TANDY 2000

Solution Systems™

335-D Washington St., Norwell, MA 02061 (617) 659-1571

*Steve McMahon's quote courtesy Suntype Publishing Systems. BYTE review by Mr. McMahon may be found in BYTE Magazine March 1985.

Circle **no. 102** on reader service card.

NEXTOPT
is used once such a parameter has been processed to advance to the next command argument.

OTHER
must follow all options; following this, one specifies what should be done if an option could not be recognized. NEXTOPT may be specified in this case, too. The unknown option itself is **argv.

After the OPT ENDOPT loop, argc contains the number of command arguments that have not yet been processed, and *argv is the first such argument. Arbitrarily many (different) options ARG can be specified. pr would be implemented approximately as follows:

```
MAIN
{
    do
    {
        OPT
            ARG 'h':
                PARM
                header = *argv;
                NEXTOPT
            ARG 'w':
                PARM
                width = atoi(*argv);
                NEXTOPT
            ARG 'l':
                PARM
                length = atoi(*argv);
                NEXTOPT
            ARG 't':
                tflag = 1;
            ARG 's':
                PARM
                delimiter = **argv;
                ++*argv;
                NEXTOPT
            ARG 'm':
                mflag = 1;
            OTHER
                if (isdigit(**argv))
                    columns = atoi(*argv),
                        NEXTOPT
                else
                    USAGE, exit(1);
        ENDOPT

        if (argc)
        {
            if (**argv == '+')
            {
                PARM
                first_page = atoi(*argv);
                continue;
            }

            dopr(*argv);
        }
        else
            dopr("-");
    } while (argc > 1);
}
```

There is a blemish: - columns must be secified as a single argument (because - alone refers to standard input).

Notes

[1] This technique was developed for the book *Introduction to Compiler Construction* by A. T. Schreiner and H. G. Friedman, Jr., Prentice-Hall, 1985.

[2] The technique requires that a pointer to a character string be returned in place of an int value. This is not possible across all implementations of C; e.g., it is probably not allowed on the 7300 systems. We guard against a portability problem by using assert( ). **DDJ**

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 192.**

```
#include <stdio.h>

#define show(x) printf("x = %d\n", x)
#define USAGE fputs("cmd [-f] [-v  #]\n", stderr), exit(1)

main(argc, argv)
int argc;
char **argv;
{
    int f = 0, v = 0;

    while (--argc > 0 && **++argv == '-')
    {
        switch (*++*argv)
        {
            case 0:                                    /* - */
```

Listing One

(Continued on page 50)

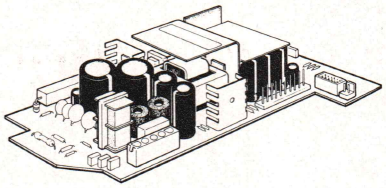
ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

Switching Power Supply!

+ 5VDC - 8 Amps
+12VDC - 5 Amps
-12VDC - 1 Amp
(81 WATTS MAX)



$29.95 ea.

4 FOR $99

ADD $2 EA. UPS

BRAND NEW UNITS, MFG. BY BOSCHERT FOR HEWLETT PACKARD! PERFECT FOR SMALL COMPUTER AND DISK DRIVE APPLICATIONS #XL81-5630. INPUT 110V/220V, 50/60 HZ. NOMINAL OUTPUTS: +5VDC @ 8A, +12VDC @ 5A, -12VDC @ 1A. TOTAL MAX OUTPUT. MUST BE LIMITED TO 81 WATTS TOTAL! (WITH PIN OUT SHEET.) ORIGINAL FACTORY BOXED.

64K S100 STATIC RAM

$139.00 KIT

NEW!

LOW POWER!

150 NS ADD $10

BLANK PC BOARD WITH DOCUMENTATION $49.95

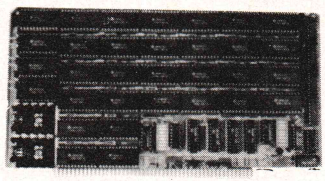
SUPPORT ICs + CAPS $17.50

FULL SOCKET SET $14.50

FULLY SUPPORTS THE NEW IEEE 696 S100 STANDARD (AS PROPOSED)

FOR 56K KIT $125

ASSEMBLED AND TESTED ADD $50



PRICE CUT!

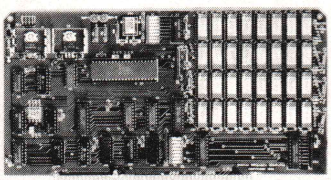
FEATURES:

* Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
* Fully supports IEEE 696 24 BIT Extended Addressing.
* 64K draws only approximately 500 MA.
* 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
* SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
* 2716 EPROMs may be installed in any of top 48K.
* Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
* Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
* BOARD may be partially populated as 56K.

256K S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

PRICE CUT!



FEATURES:

* 256K on board, using + 5V 64K DRAMS.
* Uses new Intel 8203-1 LSI Memory Controller.
* Requires only 4 Dip Switch Selectable I/O Ports.
* Runs on 8080 or Z80 S100 machines.
* Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
* Provisions for Battery back-up.
* Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
* The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
* Compare our price! You could pay up to 3 times as much for similar boards.

BLANK PCB (WITH CP/M* 2.2 PATCHES AND INSTALL PROGRAM ON DISKETTE) $69.95 (8203-1 INTEL $29.95)

(ADD $50 FOR A&T)

$169.00

#LS-100 (FULL 256K KIT)

64K SS-50 STATIC RAM

$119.00 (48K KIT)

NEW!

LOW POWER!

RAM OR EPROM!

BLANK PC BOARD WITH DOCUMENTATION $52

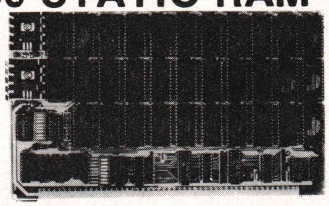
SUPPORT ICs + CAPS $18.00

FULL SOCKET SET $15.00

56K Kit $129

64K Kit $139

ASSEMBLED AND TESTED ADD $50



FEATURES:

* Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
* Fully supports Extended Addressing.
* 64K draws only approximately 500 MA.
* 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
* Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
* 2716 EPROMs may be installed anywhere on Board.
* Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
* One Board supports both RAM and EPROM.
* RAM supports 2MHZ operation at no extra charge!
* Board may be partially populated in 16K increments.

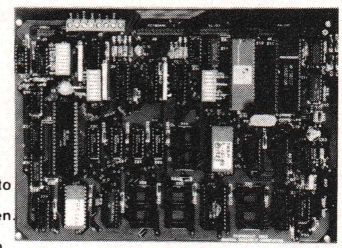
THE NEW ZRT-80

CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

* Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
* RS232 at 16 BAUD Rates from 75 to 19,200.
* 24 x 80 standard format (60 Hz).
* Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
* Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
* Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
* 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
* Composite or Split Video.
* Any polarity of video or sync.
* Inverse Video Capability.
* Small Size: 6.5 x 9 inches.
* Upper & lower case with descenders.
* 7 x 9 Character Matrix.
* Requires Par. ASCII keyboard.



BLANK PCB WITH 2716 CHAR. ROM, 2732 MON. ROM $49.95

SOURCE DISKETTE - ADD $10

SET OF 2 CRYSTALS - ADD $7.50

WITH 8 IN. SOURCE DISK! (CP/M COMPATIBLE)

$99.95 # ZRT-80

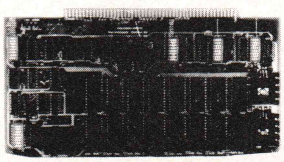
(COMPLETE KIT, 2K VIDEO RAM)

32K S100 EPROM/STATIC RAM

NEW! FOUR FUNCTION BOARD! **NEW!**

EPROM II FULL EPROM KIT $69.95

A&T EPROM ADD $35.00



BLANK PC BOARD WITH DATA $39.95

SUPPORT IC'S PLUS CAPS $16

FULL SOCKET SET $15

We took our very popular 32K S100 EPROM Card and added additional logic to create a more versatile EPROM/RAM Board.

FEATURES:

* This one board can be used in any one of four ways:
  - A. As a 32K 2716 EPROM Board
  - B. As a 32K 2732 EPROM Board (Using Every Other Socket)
  - C. As a mixed 32K 2716 EPROM/2K x 8 RAM Board
  - D. As a 32K Static RAM Board
* Uses New 2K x 8 (TMM2016 or HM6116) RAM's
* Fully Supports IEEE 696 Buss Standard (As Proposed)
* Supports 24 Bit Extended Adressing
* 200 NS (FAST!) RAM'S are standard on the RAM Kit
* Supports both Cromemco and North Star Bank Select
* Supports Phantom
* On Board wait State Generator
* Every 2K Block may be disabled
* **Addressed as two separate 16K Blocks on any 64K Boundary**
* Perfect for MP/M* Systems
* RAM Kit is very low power (300 MA typical)

PRICES SLASHED!

32K STATIC RAM KIT — $109.95

For RAM Kit A&T — Add $40

Digital Research Computers

P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 225-2309

TERMS: Add $3.00 postage. We pay balance. Orders under $15 add 75¢ handling. No. C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for for insurance.

*TM OF DIGITAL RESEARCH INC. (CALIF.)

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH INC. (CALIF.) THE SUPPLIERS OF CPM SOFTWARE

(Continued from page 48)

```
                --*argv;
                break;
            case '-':
                if (! (*argv) [1])                          /* -- */
                {
                    ++ argv, -- argc;
                    break;
                }
            default:
                do
                {
                    switch (**argv)
                    {
                        case 'f':                           /* -f */
                            ++ f;
                            continue;
                        case 'v':
                            if (*++*argv)
                                ;                           /* -v# */
                            else if (-- argc > 0)
                                ++argv;                     /* -v # */
                            else
                                break;
                            v = atoi(*argv);
                            *argv += strlen(*argv) -1;
                            continue;
                    }
                    USAGE;
                } while (*++*argv);
                continue;
            }
            break;
        }
        show(f), show(v), show(argc);
        if (argc)
            puts(*argv);
    }
```

Listing One

```
#define MAIN    main(argc, argv)                    \
                int argc;                           \
                char ** argv;

#define OPT     while (-- argc > 0 && \ **++argv    \
                    == '-')                         \
                {                                   \
                    switch (*++*argv)               \
                    {                               \
                        case 0:                     \
                            --*argv;                \
                            break;                  \
                        case '-':                   \
                            if (! (*argv) [1] )     \
                            {                       \
                                ++ argv, -- argc;\
                                break;              \
                            }                       \
                        default:                    \
                            do                      \
                            {                       \
                                                    \
                                switch (**argv)     \
                                {                   \

#define ARG     continue;                           \
        case
#define OTHER   continue;                           \
                }
#define ENDOPT  } while(*++*argv);                  \
                continue;                           \
            }                                       \
            break;                                  \
        }

#define PARM    if (*++*argv)                       \
                    ;                               \
                else if (-- argc > 0 )              \
                    ++argv;                         \
                else                                \
                    break;

#define NEXTOPT *argv += strlen(*argv)-1;
```

Listing Two

Electronic Circuit Analysis

- New release
- Transient, AC, DC analysis
- Full nonlinear
- Over 200 nodes
- Full editing
- Macro circuits
- Worst case, Monte-Carlo
- Temperature effects
- Frequency dependent parts
- Time dependent parts

For MS-DOS. 192k minimum.
$395.00

Tatum Labs
33 Main Street
Newtown, CT 06470
(203) 426-2184

Circle **no. 82** on reader service card.

"C/80 . . . the best software buy in America!" —MICROSYSTEMS

Now available in MS-DOS

Other technically respected publications like *Byte* and *Dr. Dobb's* have similar praise for **The Software Toolworks' $49.95** full featured **'C' compiler** for CP/M® and HDOS with:

- I/O redirection
- command line expansion
- execution trace and profile
- initializers
- Macro-80 compatability
- ROMable code
- and much more!

"We bought and evaluated over $1500 worth of 'C' compilers . . . C/80 is the one we use."

— Dr. Bruce E. Wampler
Aspen Software
author of *"Grammatik"*

The optional **C/80 MATHPAK** adds 32-bit floats and longs to the C/80 3.0 compiler. Includes I/O and transcendental function library all for only **$29.95!**

C/80 is only one of 41 great programs each **under sixty bucks.** Includes: LISP, Ratfor, assemblers and over 30 other CP/M® and MSDOS programs.

For your **free** catalog contact:
The Software Toolworks
15233 Ventura Blvd., Suite 1118,
Sherman Oaks, CA 91403 or call 818/986-4885 today!

CP/M is a registered trademark of Digital Research.

Circle **no. 61** on reader service card.

EXTRA!... VALUE and PERFORMANCE with Relocatable Z-80 Macro Assembler *FROM MITEK*

It's a real bargain! Here's why:

- Only $49.95 plus shipping
- 8080 to Z-80 Source Code Converter
- Generates Microsoft compatible REL files or INTEL compatible hex files
- Compatible with Digital Research macro assemblers MAC & RMAC
- Generates Ditigal Research compatible SYM files
- Full Zilog mnemonics
- INCLUDE and MACLIB files
- Conditional assembly
- Phase/dephase
- Separate data, program, common and absolute program spaces
- Customize the Macro Assembler to your requirements with installation program
- Cross-reference Generation
- Z-80 Linker and Library Manager for Microsoft compatible REL files available as a total package with Macro Assembler for only $95.00 plus shipping
- Manual only is $15

NEW for Turbo Pascal Users

Mitek's Relocatable Z-80 Macro Assembler will also:

- Generate Turbo Pascal in-line machine code include files
- Include files provide Turbo Pascal compatible machine code with assembly language mnemonics as comments

TO ORDER, CALL TOLL FREE: 1-800-367-5134, ext. 804
For information or technical assistance: 1-808-623-6361

Specify desired 5¼" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed. Include $5 for postage and handling.

Z-80 is a trademark of Zilog, Inc. MAC, RMAC, and ZSID are trademarks of Digital Research, Inc.

MITEK
P.O. Box 2151,
Honolulu, HI 96805

Circle **no. 66** on reader service card.

Information Age Issues

by Dean Gengle

The real issues in telecommunications dwarf questions about protocol and baud rate. This is an essay on responsibility.

My first experiences with a modem, a microcomputer, a television set, and a telephone linked together—with me in the loop—were much like my first discoveries regarding the true significance of portions of my own anatomy. These experiences combined the joys of intellectual insight, emotional pleasure, physical involvement, and all the wonder we associate with the notion of "surprise."

On-Line Life

I conducted my early experiments in on-line living with no misgivings at all. I was able to consider only the positive possibilities contained in this new technical capability. Modemocracy! Distributed networks! Grass roots anticipatory guidance systems! It all seemed within reach if only everyone would use these tools. The sheer weight of the new experiences themselves bolstered my initial faith that folks would come to use these tools in the "right" way.

I remember distinctly the moment when the inevitable mental disappointment occurred and reality set in—about six weeks after my first modem fix (ca 1980). I was logged on a local bulletin board system (BBS). I had just finished entering a private reply to a particular message; it was meant to go directly to another user of the system. On my screen flashed "Hi." Weird response. Commands issued to the remote computer no longer worked. Instead, the system operator (sysop) had taken over and was typing messages at me directly.

"I've been watching your session," he wrote, in the most casual and matter-of-fact manner. "Thought I'd say hello." In that instant, the scales fell from my eyes, so to speak. My illusion of privacy was shattered, an illusion maintained without my awareness by a number of supporting factors. I was working out of my own apartment, a bedroom with a spare corner space, in fact. What could be more personal or private? Also, I've never had any reason to think that my phone was tapped, so I've always regarded phone conversations as "private" My modem is hooked to the phone, ergo . . . . Besides, the remote computer system itself reinforced my illusion by allowing "private" messages between individuals and by requiring a password to read "mail."

All along, that sysop might have been looking over my shoulder, reading what I typed, aware of when and how I used the system. The implications of that made me decidedly uncomfortable. Since then, I have become aware of other sysops who've turned their computers into dictatorchips. They not only eavesdrop at will but run their systems with the firm style of a Khomeni. (Confess or I pull out your passwords!) Large systems and small are run this way. The truly astounding thing is that some people seem to *like* that sort of relationship. Still, I hope, there numbers are not great. Not here. Not in the U.S. of A.

Dean Gengle, 1150 Bryant St., San Francisco, CA 94103 (415) 861-8733.

Info Liberties

The net result of this is that I've been led, willy-nilly, into what has become an ongoing interest in civil liberties and technology. I have found that specific design decisions of an apparently "purely technical" nature can and often do have an effect on what we consider to be our personal liberties and freedoms. Conversely, social values can affect what and how new technology is designed.

The controversies that arise from this interplay taken together can be called *information age issues*. They all focus on the handling of information: who generates it; who can store it; who can ship it around; who can have it; who can control it; who can copy it and who cannot; what is and is not legitimate information in a given instance—hence, "information age."

It is becoming increasingly difficult to draw a firm line between the idea of telecommunications and the idea of computing. Some commentators have coined terms such as *telecomputing* and *compunications* to make this point. (I will spare the reader any further use of these, however.) Distinctions between media are blurring fast. All kinds of information now can take digital form to be zapped across various channels, whether phone lines, fiber optic cables, cellular networks, or satellite transponders.

For the time being, I find it useful to think of *telecommunications* as any form of electronic transmission requiring temporary or permanent physical connections.

The examples that follow are representative of the dilemmas we already confront. I can only hint at telecommunications-related conundrums (or nightmares) we may have to face in the months and years directly ahead.

Freedom from Eavesdropping

The kind of surveillance conducted by the sysop I described earlier is not the only kind that new technologies have made possible. Signals of all kinds are now being intercepted on a widespread scale. On the governmental level, spy agencies around the world engage in SIGINT (signal intelligence) work:

- Radio and television broadcasts, voice and data phone calls, satellite channels, international telex and TWX and Western Union transmissions, and amateur radio broadcasts are all monitored. The superpowers conduct the most sophisticated monitoring, as one would expect, but the technology of signal interception is getting cheaper every day.
- Telephone company officials, FBI agents, and credit card companies go on "fishing expeditions" in local microcomputer-based message systems. Ostensibly, they are looking for exchanges of "illegal" data, such as computer access numbers and passwords, phone credit card numbers, and so on. Who knows what other subversive material these official intruders might turn up?
- Although laws surrounding the tapping of voice telephone calls are stringent when they deal with what may and may not be used as trial evidence or when warrants are required and when they are not, no such laws cover eavesdropping on private data transmissions. With large computers doing the listening and sifting, it is a simple matter to monitor thousands of data phone calls.

Meanwhile, in the commercial sector, cellular mobile telephone systems are being marketed heavily in major metropolitan areas across the country. However, cellular channels can be monitored with easily built "scanners" similar to those used to monitor police and CB channels.

Thus, as the cellular service suppliers themselves admit, those top level business calls we make from the middle of traffic jams may not be private. The service suppliers downplay this aspect of cellular mobile telecommunications. After all, there are 666 assigned channels, and the channel assignment for a particular call *does* shift as the user moves from cell to cell. But even if it isn't technically feasible to build a scanner capable of following a particular call, you probably shouldn't conduct sensitive business by cellular phone anyway, according to the suppliers. So much for productivity in traffic jams.

A few companies already are planning to connect modems and micros to cellular telephones. Get your electronic mail from anywhere, anytime. There is no reason to think this evolution will stop until someone makes a device that is a cellular phone, voice and data terminal, and microcomputer, all in one portable package. The temptation to use such a device will grow right along with the temptation to listen in.

"Freedom from eavesdropping" is more apt as a rallying cry than "right to privacy" because certain forms of eavesdropping do not intrude directly on our privacy. For example, others can gather and analyze information about your phone calls even if they do not monitor directly the content of the calls themselves. Within many corporate settings, this already is being done.

Information about whom you call, when you call, the duration of the call, and the frequency of calls to particular numbers all can be collected without your knowledge. To some extent, the same laws covering voice phone tapping hedge the ability of police and government to use such information. However, no such laws cover intraorganizational phone monitoring. Monitoring systems are marketed under the rubric of "phone management" and "automatic call distribution and reporting systems."

One company advertisement for this kind of system is a touch Orwellian in its copy: "To check out your staff's performance, you can either walk around the office. Or get one of [our] automatic call distribution systems. They actually print out detailed reports on every single person's hourly performance: Who needs a raise. And who needs improvement."

Freedom to Play vs. the Drive for Security

Given the complexity of our society, we are far more likely to lose our valued freedoms bit by bit than all at once. Freedoms also are more likely to erode faster as a result of nongovernmental institutional decisions.

As technicians, software designers, hardware designers, and computer users, we have as much to lose through the misapplication of technology as anyone else. We are

also, perhaps, in a better position to prevent misapplications than anyone else.

There is much attention paid these days to the ubiquitous "hacker," whoever that may be. The hacker is linked in the public mind with computer crime. This was not always so, although the reasons for this association are many. Yet, whether one thinks of oneself as a hacker or not, the solutions to the problems of computer crime don't seem to lie in the direction of hiring more consultants. Nor does it make much sense to try to rehabilitate the image of the hacker by making him or her into some sort of cultural hero(ine).

Along with the hacker image, the public is being sold a bill of goods concerning computer crime. Such crime, by and large, is portrayed as something an individual does to an organization, using a microcomputer. The public sees computer crime as "breaking and entering" large systems, whether to take money gains, information gains, or both. But other kinds of computer abuse are not labelled as "crimes" or "criminal" and, for the most part, are not even mentioned in the mass media that shape public consciousness on such matters. The daily papers do not discuss the idea that data eavesdropping by government, corporate, or even private individuals on private citizens may also be criminal. Nor does the idea that the IRS or NSA may be tapping the communications and computers of private individuals receive much consideration. The laws that are discussed and passed are laws aimed at individual abuse, not large-scale institutionalized abuse.

Often, the quest for security tramples our rights to privacy and individuality. In the drive for absolute security in an absolutely efficient society, we may lose the one thing we need in order to survive and prosper in the global information economy: the freedom to play and its attendant creativity.

No matter what the end product, creative (and therefore innovative) ideas are nurtured in an atmosphere of play. The balance between control and freedom must be struck in actual design decisions we make from day to day. The impact of one implementation vs. another on our overall freedoms must be part of design decisions.

High on the list of needed design developments in an era of mass telecommunications is some sort of public key encryption and signature system that is mass-producible and inexpensive and secure from any kind of eavesdropping. The jury still seems to be out on whether or not the DES (Data Encryption Standard) is really secure from megacomputer cracking power. This is an area where some effective designs could improve all our on-line lives, but it is only one example.

The Future of Paranoia

Future problems in telecommunications may make today's problems seem relatively trivial by comparison. As expert systems take up residence in our telecommunications networks, designers will have to become even more concerned with individual freedoms—constitutional and civil liberties—than they are now.

For example, it is feasible to expect that various information suppliers, data banks, and mass information utilities will be able to learn about users in sophisticated ways: what your interests are, what items to call to your attention based on your past use of the system, what you read and what you ignore, what kind of equipment you have, your age, marital status, and economic position, and so forth.

It would be nice to interact with a system that appears custom tailored to your needs and use. However, it is not a big leap to go from information about how you've interacted with, say, a "smart Source" or "intelligent CompuServe" to a psychological and/or political profile. This sort of information analysis is good not only for marketers who want to sell you things but for politicians, power junkies, and competitors who want to know all they can about you for reasons of their own.

Approaches to Action

People directly involved in making technical decisions for design or implementation are, as I said earlier, in a good position to act as *ad hoc* advisers to the general public, the media, and lawmakers at local, state, and federal levels. Technical people, along with artists, can act as "antennae for the race" in matters of sociotechnical importance. If you are involved in the early design stages of an expert software system to correlate IRS data with SSI information, to take one hypothetical example, you'll know before the rest of us do how that system can be abused and put to uses never intended in the first place.

We've all heard that the price of liberty is eternal vigilance. But just how one remains vigilant in an age of rapid technological change is not so obvious. Responses to upcoming issues can take a variety of forms, whether individual, institutional, or governmental. It has become an article of faith for me, having studied and worked in this arena for several years, that individual responses are preferable to institutional ones and institutional ones preferable to governmental intervention. However, we will need all these approaches.

Individual Responses

Individual technicians and designers play a large role in the creation of technical standards and on standards bodies. In telecommunications, for instance, global standards for videotext, viewdata, and teletext systems have yet to emerge. Standards can affect the kinds of competition possible and hence the marketplace. It is not unheard of for standards to be adopted, not on their merit, but for overtly political reasons.

More individual participation in the creation of new standards, therefore, is one way in which technicians and designers can respond.

Educating our friends and neighbors who are not technically inclined is another way to respond to the changes raining down. Explicit information on the connections between technical possibilities and quality of life is needed on a wider scale than ever before.

Organizational Responses

When biotechnology began to hint that doctors and biological technicians could have a profound influence on the

direction and quality of human life, medical schools and other establishments of higher learning began to teach formal courses in medical and biological ethics.

Today, when computer technology has had—is having—an equally profound effect on our quality of life, almost none of the establishments responsible for technical education are providing courses equivalent to those taken by medical students. A few formal studies of "technical ethics" and "information age ethics" are being conducted, but these are one-shot projects, not long-term educational efforts. Therefore, one possible institutional response to information age issues is to provide courses that combine the study of ethics and morals with the study of information science and the future of telecommunications.

Legal Responses

It's important that, in the discussion of what is and is not "computer crime," we not let high-priced "security consultants" or politicians with their own axes to grind make our laws for us.

It is especially important that we not let the rare occurence of a "hacker break-in" distract us from the larger and potentially more serious issues at hand. The Privacy Act and the Freedom of Information Act, for example, have suffered consistent erosion over the last 10 years by administrative decree and bureaucratic red tape.

As more government information is put on line and made accessible by computer, no one is talking about our rights to access information that is *not* private and that was collected legally at taxpayers' expense. Census data and Library of Congress information come to mind. (You can get census data now, certainly, if you have certain kinds of computers and are willing to pay a lot for it.) The government is probably the largest single source of useful information in this country. Making that information available to the micro-using public is an issue worthy of Congressional attention.

Last Words

I don't have any. I am willing to bet that some of the people who read this piece are engaged, even now, in debates on design that will come to wider attention only after a particular piece of outrageous hardware or software has been manufactured and distributed. Even now, you probably can think of a dozen or so issues I've not been able to cover here because of my own ignorance or lack of space.

In this business of telecommunications, there are no last words. Therefore, for all our sakes, I urge you to add yours to the heap.

Dean Gengle is a software documentation specialist and consultant at CommuniTree Group in San Francisco. He is the author of The Netweaver's Sourcebook. **DDJ**

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 193.**

TAKE THE PAIN OUT OF BACKUP

Famous Excuses For Not Doing Backups

Backing up a Winchester onto multiple floppies takes forever.

PIP makes you remember what you changed — too much trouble.

Disk failures only happen to the other guy.

Floppy-To-Floppy Incremental Backup For Only $40.

Like PIP but copies just the files that have changed.

Order Qbax1.

The Remedy: Copy Just What's New — Automatically!

Qbax2 will:
Split files bigger than one floppy.
Track all the pieces.
Tell you which floppy it needs and when.
Give you a built-in catalogue.
Reclaim wasted floppy space.

Qbax2 only $95 from:

Amanuensis, Inc.
R.D. 1, Box 236
Grindstone, PA 15442
(412) 785-2806

For CP/M 2.2 on 8" SSSD and popular 5¼" formats. Shipping $2 U.S. and Canada, $4 overseas. Qbax TM Amanuensis, Inc. CP/M® Digital Research.

FOR EVERYONE WITHOUT A DISPENSATION FROM MURPHY'S LAW

Circle **no. 5** on reader service card.

Modems: 2400 Bit/Sec and Beyond

by Dale Walsh

Electronic data communication, in its simplest form, employs binary signaling common in logic and microprocessor circuits. A "high" signal is one, and a "low" signal is zero. We have seen dramatic increases in how fast electronic logic devices can switch binary logic levels. Each new computer, it seems, is several times faster than its predecessor. This is achieved primarily by reducing large-scale integration (LSI) devices in size to increase device speeds. We in the communications field would say "they kept binary signaling but increased signal bandwidth."

The one sure thing about the new 2400-baud modems is that we'll find them intolerably slow before the next thing comes along.

We usually associate bandwidth with frequency limits: our hearing bandwidth is from 20 Hz to 12,000 Hz, for example. But bandwidth also limits how fast things can move. Run in water and you appreciate that water is a narrow bandwidth medium, at least for physical movement.

In their efforts to keep pace with the demand for higher speeds, modem designers immediately collided with a limited resource: by and large, signal bandwidth, set by the telephone channel, is fixed. Two approaches will increase speed: (1) use more of the available bandwidth, and (2) construct densely packed signaling alphabets so that each signaling interval uniquely specifies more than one data bit.

Dale M. Walsh, U.S. Robotics Inc., 477 East Butterfield Rd., Lombard, IL 60148.

Available Bandwidth

The available telephone channel bandwidth is approximately 200–3200 Hz. Lower speed modems carve out narrow slices near the center where channel distortions are at a minimum. But as you move out from the middle, things get sticky. Band-edge characteristics vary widely from call to call and, near the limits, are extremely hostile to data signals. Modem evolution, as a result, favored development of complex signal alphabets before the final assault on unused bandwidth.

The Bell 103 and other dial-up, full duplex modems belong to a class called Frequency Division Multiplexing (FDM) modems. They send and receive simultaneously using two non-overlapping frequency bands—much like two radio stations tuned at different frequencies. By protocol, the calling modem uses the lower frequency band; hence the terms "Originate" and "Answer" bands. The figure (page 57) shows these bands for the Bell 103 and Bell 212 modems.

Signal States

"One if by land, two if by sea," and Paul Revere was on his way. The flash of light was a data symbol, and one bit per data symbol did the job.

The 300 bit/sec 103 modem uses a similar scheme: one tone for a one and a different tone for zero. It's a fairly simple scheme and easily implemented with today's technology. The Originate and Answer bands each use about 300 Hz of the roughly 3,000 Hz bandwidth that is available.

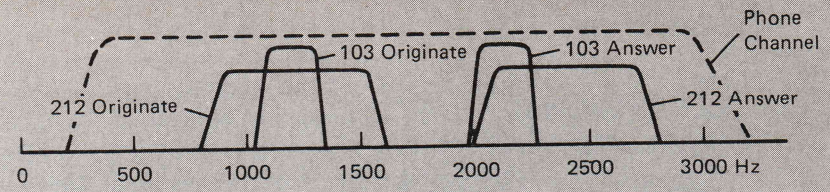
To quadruple data rates to 1200 bit/sec, modem designers made three changes. First, they changed the basic signaling scheme from frequency modulation (as in FM radio) to phase modulation; the latter more efficiently uses available bandwidth. Second, they increased the signaling alphabet from two states (0 and 1) to four states (0, 1, 2, 3); hence, each data symbol uniquely signals two bits. Finally, they doubled the baud rate (the rate at which data symbols are sent from transmitter to receiver) from 300 to 600; this boosted the Originate and Answer bandwidths to approximately 800 Hz each.

These changes significantly increased modem complexity. But LSI technology rode to the rescue. Faster microprocessors handled the modulation and demodulation tasks. More importantly, the complicated 212 analog filters that separate high and low (Answer and Originate) bands were reduced to single LSI chips.

2400

The move from 1200 to 2400 bit/sec distinctly ups the ante.[1] Designers decided not to increase signal bandwidth further, primarily so that they could use 212 filter chips in 2400 bit/sec modems and so that 2400 modems could conveniently fall back to a 212 mode.

Instead, they increased speed by changing the signaling scheme from phase modulation to quadrature amplitude modulation (QAM). In QAM, you can increase the signal alphabet by varying both signal phase and amplitude. A 2400 bit/sec modem sends 16 phase-amplitude combinations. Each combination corresponds to four data bits. Therefore, the modem baud rate between transmitter and receiver is the same for both 212 and 2400, but each 212 data symbol carries two data bits while each 2400 modem data symbol carries four data bits.



SIGNAL BANDWIDTHS

Figure

New Release

C + UTILITY LIBRARY = PRODUCT

• We have over 300 *complete, tested, and, documented functions.* All source code and demo programs are included.

• The library was specifically designed for software development on the IBM PC, XT, AT and compatibles. There are *no royalties.*

• Over 95% of the source code is written in C. *Experienced programmers* can easily "customize" functions. *Novices* can learn from the thorough comments.

We already have the functions you are about to write

Concentrate on software development—not writing functions.

THE C UTILITY LIBRARY includes:

• Best Screen Handling Available • Windows • Full Set of Color Graphics Functions • Better String Handling Than Basic • DOS Directory and File Management • Execute Programs, DOS Commands and Batch Files • Complete Keyboard Control • Extensive Time Date Processing • Polled ASYNC Communications • General DOS/BIOS gate • Data Entry • And More •

• The Library is compatible with: Lattice, Microsoft, Computer Innovations, Mark Williams and DeSmet. Available Soon: Digital Research, Aztec and Wizard.

C Compilers: Lattice C—$349, Computer Innovations C86—$329; Mark Williams C—$449.

C UTILITY LIBRARY $185. Special prices on library & compiler packages.

Order direct or through your dealer. *Specify compiler* when ordering. Add $4.00 shipping for UPS ground, $7.00 for UPS 2-day service. NJ residents add 6% sales tax. Master Card, Visa, check or P.O.

ESI

ESSENTIAL SOFTWARE, INC

P.O. Box 1003 Maplewood, New Jersey 07040 914 762-6605

Circle no. 36 on reader service card.

BORED?...

...waiting for C programs to compile and link?

Use **C-terp**
the **complete** C interpreter

This is the product you've been waiting (and waiting) for!

Increase your productivity and avoid agonizing waits. Get instant feedback of your C programs for debugging and rapid prototyping. Then use your compiler for what it does best...compiling efficient code ...slowly.

C-terp Features

- Full K&R C (no compromises)
- Complete built-in screen editor-- no half-way house, this editor has everything you need such as multi-files, inter-file move and copy, global searching, auto-indent, tab control, and much more.
- Fast--Linking and semi-compilation are breath-takingly fast. (From edit to run completion in a fraction of a second for small programs.)
- Convenient--Compiling and running are only a key-stroke or two away. Errors direct you back to the editor with the cursor set to the trouble spot.
- Object Module Support – Access functions and externals in object modules produced by C86 or Lattice C or assembly language. Utilize your existing libraries unchanged!
- Complete Multiple Module Support– Instant global searches, auto-compile everything that's changed, etc.
- Many more features including batch mode, 8087 support and symbolic debugging.
- Runs on IBM PC, DOS 2.x, 192K and up.
- **Price: $300.00 (Demo $45.00) MC, VISA**

Price of demo includes documentation and shipping within U.S. PA residents add 6% sales tax. Specify C86 or Lattice version.

GIMPEL SOFTWARE
3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261

*Trademarks: C86 *(Computer Innovations)*, Lattice *(Lattice Inc.)*, IBM *(IBM Corp.)*, C-terp *(Gimpel Software)*

Performance

What does this cost? First, design complexity quadruples. The design is more exacting and permits fewer simplifying short cuts.

Second, an inherent performance loss is associated with using 16 different signals rather than four. Imagine a signal system that flashes one of four colors, each color representing two bits. To convert to 16 colors would degrade the system's ability to distinguish colors, especially when they are surrounded by other interfering color flashes.

However, the performance difference between 212 and 2400 modems is not as great as one might expect. The 2400 modems use adaptive equalizers that "learn" telephone channel characteristics, compensating for call-to-call variations. An adaptive equalizer is a little like an auto-focusing camera device that also balances the color spectrum and corrects lens imperfections. Because a sharply focused receiver signal is more noise tolerant, the 2400 receiver can achieve 80–90 percent of its potential noise immunity on every call. Most 212 designs use a compromise equalizer ("one size fits all") approach that sacrifices performance for design simplicity. This difference significantly narrows the performance gap between the two modems.

But, more to the point, how well the 2400 modem works depends on its performance potential compared to the performance limits imposed by the U.S. telephone network. After reading a recent survey of signal levels, noise, and distortions on the U.S. network, one can conclude that well-designed 2400 modems will perform successfully on a very high percentage of such connections. Indeed, this is borne out by units already in the field: a limited test using a U.S. Robotics Courier 2400 modem on long distance connections yielded 95 percent successful calls.

Higher Speeds

Doubling speed from 2400 to 4800 is not as straightforward as going from 1200 to 2400 because 2400 bit/sec is the maximum practical speed for FDM dial-up modems (like the Courier 2400). There are two barriers. First, network changes will make extending the signal alphabet beyond 16 states highly questionable. Second, the remaining option (to use more of the available signal bandwidth) chews up more bandwidth than is available! At 4800 and above, the Answer and Originate bands overlap, and design complexity escalates dramatically.

It *is* possible to mix Transmit and Receive signals in the same frequency band. Each end must ignore its own transmitter and pick up the remote transmitter. This is called echo cancellation. Echo cancellation and adaptive equalization (mentioned above) are similar in that they both "learn" line conditions and adaptively compensate. But echo cancellation requires more computations with about twice as many bits of arithmetic. Several European modems successfully use echo cancellation, but they are fairly expensive compared to 2400 bit/sec modems.

CCITT recommendation V.32 defines a new dial-up 9600 bit/sec modem that includes a 4800 bit/sec mode. The Originate and Answer bands completely overlap, and each occupies 90 percent of the available bandwidth. Originally, the modem was conceived for local access to digital data networks. However, added features now allow long haul operation on international circuits and in large countries like the U.S.

Recommendation V.32 is very new and subject to change as the first models are introduced. Measured by computations per second and bits of resolution, these modems are roughly 64 times more complex than 2400 bit/sec modems. For this reason, it will be quite some time before V.32 modems can compete in the high-volume modem market.

Notes

[1] The discussion here concerns dial-up, full duplex modems in the 103 and 212 class. Leased-line modems typically operate 4,800–14,400 bit/sec.

DDJ

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 197.**

THE PROGRAMMER'S SHOP™

helps compare, evaluate, find products. Straight answers for serious programmers.

SERVICES

- Programmer's Referral List
- Compare Products
- Help find a Publisher
- Evaluation Literature free
- BULLETIN BOARD - 7 PM to 7 AM 617-826-4086
- Dealer's Inquire
- Newsletter
- Rush Order
- Over 700 products

Free Literature - Compare Products

Evaluate products Compare competitors. Learn about new alternatives. One *free call* brings information on just about any programming need. Ask for any ***"Packet"*** or "Addon Packet": ☐ ADA, Modula ☐ "AI" ☐ BASIC ☐ "C" ☐ COBOL ☐ Editors ☐ FORTH ☐ FORTRAN ☐ PASCAL ☐ UNIX/PC or ☐ Debuggers, Linkers, etc.

RECENT DISCOVERIES

SMALL TALK for PCDOS - "Methods" has objects, windows, browser, inspector. PCDOS $239

ARTIFICIAL INTELLIGENCE

EXSYS - Expert System building tool. Full RAM, Probability, Why, Intriguing, serious. PCDOS $275

GC LISP - "COMMON LISP", Help, tutorial, co-routines, compiled functions, thorough. PCDOS Call

INSIGHT 1 - Expert Sys. Dev't, decent PCDOS $95

M Prolog - full, rich, separate work spaces. MSDOS $725

PROLOG-86 - Learn fast, Standard, tutorials, samples of Natural Language, Exp. Sys. MSDOS $125

TLC LISP - "LISP-machine"-like, all RAM, classes, turtle graphics 8087. CP/M-86, MSDOS $235

Expert System front-ends for PROLOG: APES ($275), ES/P ($895)

Other solid alternatives include: MuLISP-86 ($189), WALTZ LISP for CPM ($159), MicroPROLOG ($275)

EDITORS FOR PROGRAMMING

BRIEF Programmer's Editor - undo, windows, reconfigurable, macro programs, powerful. PCDOS $195

VEDIT - well liked, macros, buffers, CPM-80-86, MSDOS, PCDOS $119

MACINTOSH

We evaluate, carry every available programmers product. Ask.

C LANGUAGE

C Terp by Gimbel, full K & R, .OBJ, ASM interface, 8087. MSDOS $275

INSTANT C - Interactive development - Edit, Source Debug, run. Edit to Run - 3 Secs. MSDOS $495

"INTRODUCING C" - Interactive C to learn fast. 500 page tutorial, examples, graphics. PCDOS $95

MEGAMAX C - native Macintosh has fast compile, tight code, K&R, toolkit, .OBJ, DisASM MAC $275

C ADDONS

APPLICATION TOOLKIT by Shaw - Complete: ISAM, Screen, Overlay mgnt, report gen, Strings, String math. Source. CPM, MSDOS $495

COMMUNICATIONS by Greenleaf ($159) or Software horizons ($139) includes Modem7, interrupts, etc. Source. Ask for Greenleaf demo.

C SHARP Realtime Toolkit - well supported, thorough, portable, objects, state sys. Source MANY $600

PRE-C - LINT, full, enhanced, all Cs. MSDOS $345

DEBUGGERS

PERISCOPE DEBUGGER - load after "bombs", symbolic, "Reset box", 2 Screen, own 16K. PCDOS $279

SOURCE PROBE by Atron for Lattice, MS C, Pascal. Windows single step, 2 screen, log file. $395

FORTRAN LANGUAGE

DR/Fortran-77 - full ANSI 77, 8087, overlay, full RAM, big arrays, complex NUMS., CPM86, MSDOS $249

MacFORTRAN - full '77, '66 option, toolbox, debugger, 128K or 512K, ASM-out option MAC $375

Ask about Microsoft, Supersoft, others.

OTHER LANGUAGES

ASSEMBLER - ask about FASM-86 ($95), ED/ASM ($100) - both are fast, compatible, or MASM ($125), improvements.

BetterBASIC all RAM, modules, structure. BASICA - like $185

HS/FORTH - '79 & '83 Standards, full RAM, ASM, BIOS, interrupts, graph, multi-task, optimizer MSDOS $250

MBP COBOL has screen control, strong doc, '74 interm., fast. MSDOS $680

SUPPORT PRODUCTS

BASIC DEVELOPMENT SYSTEM - (BDS) for BASICA; Adds Renum, crossref, compress. PCDOS $115

CODESMITH - visual, interactive debugger. Symbolize, modify code $129

FASTER C - Lattice users eliminate Link step. Normal 27 seconds. Faster C in 13 sec. MSDOS $95

PLINK-86 for Overlays, most lang., segment control. MSDOS $325

"C" LANGUAGE

	OUR PRICE
MSDOS: C86-8087, reliable	call
Instant C - Inter., fast, full	495
Lattice C - the standard	call
Microsoft C 3.0 - new	279
Williams, debugger, fast	call
CPM80: EcoPlus C - faster, SLR	275
BDS C - solid value	125
MACINTOSH: Hippo II	375
Megamax - optimizer, full	275
Consulair's MAC C	275

Compare, evaluate, consider other Cs

BASIC

	RUNS ON	
Active Trace-debug	86/80	75
BASCOM-86 - MicroSoft	8086	279
BASIC Dev't System	PCDOS	115
BetterBASIC - 640K	PCDOS	185
CB-86 - DRI	CPM86	419
Prof. BASIC Compiler	PCDOS	89
CADSAM-Full Btree, source	MSDOS	150
SCREEN SCULPTOR	PCDOS	115

Ask about ISAM, other addons for BASIC

SERVICE

ALL PRODUCTS - We carry 700 products for MSDOS, CP/M 86, CP/M 80, Macintosh and key products for other micros.

EDITORS Programming

	RUNS ON	OUR PRICE
BRIEF - Intuitive, flexible	PCDOS	195
C Screen with source	86/80	75
Epsilon - like EMACS	PCDOS	195
FINAL WORD-for manuals	86/80	215
PMATE-powerful	8086	185
VEDIT-full, liked	86/80	119
XTC - multitasking	PCDOS	95

COBOL

Dig. Res-decent	MSDOS	500
Macintosh COBOL-Full	MAC	1850
MBP-Lev. II, native, screen	MSDOS	885
Micro Focus Prof.-Full	PCDOS	call
Microsoft-Lev II, no royal	MSDOS	500
Ryan McFarland-portable	MSDOS	695

LANGUAGE LIBRARIES

GRAPHICS: GraphiC-source in C	MSDOS	219
GRAPHMATIC-3D: FTN, PAS	PCDOS	125
HALO-fast, full-all lang.	PCDOS	139
FILE MGNT: BTrieve-all lang.	MSDOS	215
CIndex + -source, no royal.	86/80	369
CTree-source, no royal.	ALL	369
dBC ISAM by Lattice	8086	229
dB VISTA- "Network" Structure	MSDOS	465
PHACT-up under UNIX, addons	MSDOS	225
OTHER: CUtil by Essential	MSDOS	129
Greenleaf - 200 +	MSDOS	159
CSharp - **Real-Time**	MSDOS	600
PORTABLE C to PC, Mac, II	Many	125
SOFT Horizons - Blocks I	PCDOS	139
SCREEN: CURSES by Lattice	PCDOS	125
CView - input, validate	PCDOS	195
MetaWINDOW - icons, clip	PCDOS	139
PANEL - many lapg, term	MSDOS	249
ProScreen - windows, source	PCDOS	415
Windows for C	MSDOS	175

FORTRAN

	RUNS ON	OUR PRICE
MS FORTRAN-86 - Impr.	MSDOS	$ 239
DR Fortran-86 - full '77'	8086	249
PolyFORTRAN-XREF, Xtract	PCDOS	165

OTHER PRODUCTS

Assembler & Tools - DRI	8086	159
Atron Debugger for Lattice	PCDOS	395
cEnglish - dBase to C	MSDOS	750
C Helper: DIFF, xref, more	86/80	135
CODESMITH-86 - debug	PCDOS	1 9
MacASM-full, fast, tools	MAC	115
MBP Cobol-86 - fast	8086	885
Modula 2 for MAC,	PCDOS	90
Micro: SubMATH-FORTRAN full	86/80	250
Microsoft MASM-86	MSDOS	125
MSD Debugger	PCDOS	119
Multilink - Multitasking	PCDOS	265
PC FORTH + -well liked	MSDOS	219
PFIX-86 Debugger	MSDOS	169
PL.1-86	8086	495
Polylibrarian - thorough	MSDOS	95
PolyMAKE	PCDOS	95
PROFILER by DWB - flexible	MSDOS	109
Prolog-86-Learn, Experiment	MSDOS	125
SYMD debugger-symbols	PCDOS	119
TRACE86 debugger ASM	MSDOS	115

Call for a catalog, literature, and *solid value*

800-421-8006

THE PROGRAMMER'S SHOP™

128-D Rockland Street, Hanover, MA 02339

Visa Mass: 800-442-8070 or 617-826-7531 MasterCard 8517

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and **POs**. All formats available.

C UART Controller

by Don Gay

This article describes a program that allows easy modification of the parameters of a programmable USART. The program, written in C, is readily adapted to a number of different UART chips. This particular program is for the Signetics/National Semiconductor 2651 USART.

Recently, I constructed a serial port board to let me use a modem on my microcomputer. After looking at some of the designs in *Interfacing to S-100/IEEE 696 Microcomputers*[1] and checking the local availability of UART chips, I decided to implement the design on page 183 of the book, which makes use of the 2651 USART.

This USART is fully programmable with respect to baud rate, stop bits, parity, word length, and so on, and changes to my BIOS would program the USART at boot time. However, I needed more flexibility. When transferring files to an adjacent machine, I would have appreciated a simple way to change the USART parameters, namely, the baud rate, resetting it for higher rates.

SETP.C

The Listing on page 62 shows the results of my attempt at solving this problem. The program is called with a number of command line switches, which can be used to modify the port parameters. I have designed the program to run entirely from the command line to make it easily adaptable to systems that have an autoexecute on boot: anyone can boot up in whatever mode is required without modifying the BIOS.

An examination of the listing reveals four command line switches. These allow the user to change the baud rate (–b), number of stop bits (–s), parity (–p), and word length (–l). All invalid switches are ignored, as are values out of range for any of the parameters. After all command line switches have been effected, the program shows the values of all the port parameters. Trying to execute the program with no switches on the command line produces a "Usage" description, followed by the current port parameters. Here is an example of how the program typically is called:

```
A> setp –b1200 –s1 –pn –l8
```

This would configure the USART to transmit/receive at 1200 baud, 1 stop bit, no parity, and a character length of 8 bits.

A tasty little morsel from Wahroonga.

USART Programming Details

A 2651 USART uses four consecutive I/O ports:

(1) Transmit/Receive holding register
(2) Status register
(3) Mode register 1 and 2
(4) Command register

The USART parameters are controlled by writing to the mode registers. By reading the values in the mode registers, you can determine the current parameters of the USART.

There are actually two mode regis-

Don Gay, 136 Fox Valley Road, Wahroonga, 2076, N.S.W., Australia.

ters at the same I/O address, each accessed by a consecutive read or write. Read from the mode register once and you will get the contents of one of the mode registers; read from it again and you will get the contents of the other mode register. A similar situation exists for writing. After a reset, the first read/write accesses the first mode register. As long as reading and writing are done in pairs, you will always know with which register you are dealing.

The command register controls the level of *RTS and *DTR lines, as well as enabling and disabling transmit/receive. Because the values written to the command register need not change once the port has been initialized, the program does not allow alteration of the command register from the command line switches. These switches alter only the contents of the mode registers.

As is usually the case with programmable peripheral interfaces, each bit has a special significance in the controlling mode registers. Figures 1 and 2 (at right) show the importance of each bit in the mode registers of the 2651.

The USART is configured in this program in asynchronous mode with internal receive and transmit clocks. This is probably the most common mode used with ordinary devices, such as modems and terminals, so these characteristics are "hard coded" in the program. Refer to the 2651 documentation[2] to find out other ways to operate the port.

References

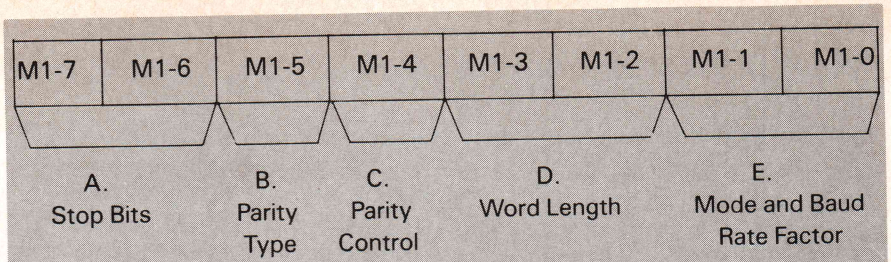
[1] Sol Libes and Mark Garetz. *Interfacing to S-100/IEEE 696 Microcomputers.* California: Osborne/McGraw-Hill, 1981.

[2] National Semiconductor Data Sheet. "INS2651 Programmable Communications Interface." October 1980.

DDJ

(Listing begins on next page)

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 195.**



A. Stop Bits
- 00 = Invalid code.
- 01 = 1 stop bit.
- 10 = 1.5 stop bits.
- 11 = 2 stop bits.

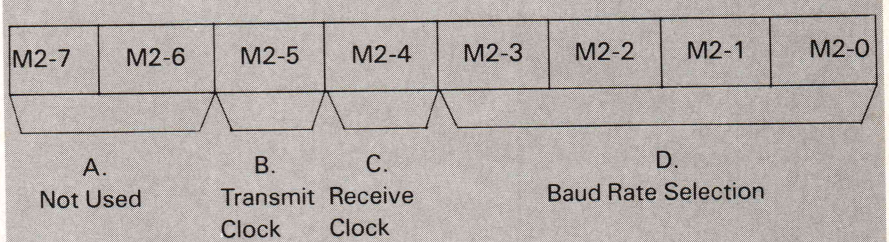
B. Parity Type
- 0 = Odd.
- 1 = Even.

C. Parity Control
- 0 = Disabled.
- 1 = Enabled.

D. Word Length
- 00 = 5 bits.
- 01 = 6 bits.
- 10 = 7 bits.
- 11 = 8 bits.

E. Mode and Baud Rate Factor
- 00 = Synchronous mode (not used).
- 01 = Synchronous mode (not used).
- 10 = Asynchronous mode 16x rate.
- 11 = Asynchronous mode 64x rate.

Figure 1
Mode Register 1 Format



A. Not Used

B. Transmit Clock
- 0 = External clock
- 1 = Internal clock

C. Receive Clock
- 0 = External clock
- 1 = Internal clock

D. Baud Rate Selection (only for async 16x mode)

0000 = 50 baud	1000 = 1800 baud
0001 = 75 baud	1001 = 2000 baud
0010 = 110 baud	1010 = 2400 baud
0011 = 134.5 baud	1011 = 3600 baud
0100 = 150 baud	1100 = 4800 baud
0101 = 300 baud	1101 = 7200 baud
0110 = 600 baud	1110 = 9600 baud
0111 = 1200 baud	1111 = 19200 baud

Figure 2
Mode Register 2 Format

C UART Controller Listing (Text begins on page 60)

```
/*************************************************************************/
/*                                                                       */
/*          Serial Port Configuration Program  (2651 PCI)                */
/*                                                                       */
/*   Source filename  : SETP.C                                           */
/*   Program Language : C/80 3.0 (Software Toolworks)                    */
/*   Program Author   : Don Gay                                          */
/*   Last Revision    : June 11, 1984                                    */
/*                                                                       */
/*************************************************************************/

#include printf.h                    /* Header for formatted output */

#define REV_DATE        "11/06/84"

#define PORT_BASE       0               /* 2651 PCI base address    */
#define MODE_REG        PORT_BASE+2     /* Mode register            */
#define CMND_REG        PORT_BASE+3     /* Command register         */

#define DISABLE         0x32            /* Disable UART recv/xmit   */
                                        /* Force *DTR and *RTS low  */
#define ENABLE          0x37            /* Enable UART recv/xmit    */

/* Store legal values for all paramters */

char *baudr[] = {"50","75","110","134.5","150","300","600",
                 "1200","1800","2000","2400","3600","4800",
                 "7200","9600","19200"},
     *sbits[] = {"0","1","1.5","2"},
     *parit[] = {"N","O","N","E"},
     *wordl[] = {"5","6","7","8"};

int baud_rate, stop_bits, parity, word_length;

main(argc,argv)
int argc;
char *argv[];
{
   printf("\nSerial port configuration program as of %s\n",REV_DATE);
   get_prm();                       /* get current port parameters */
   if (argc>1)                      /* process the command line */
      while (argc-- > 1)
         process(argv[argc]);
   else                             /* Give some help */
      show_usage();
   show_prm();                      /* Display parameters */
   set_prm();                       /* Re-program PCI     */
}

process(arg)                        /* Try to match the command line */
char *arg;                          /* argument with a valid switch  */
{                                   /* If valid, set up new value    */
   int n;
   if (*arg == '-') {
      switch(toupper(*++arg)) {
         case 'B': n=match(++arg,baudr,16);
                   if (n>=0) baud_rate=n;
                   else error("baud rate value",arg);
                   break;
         case 'S': n=match(++arg,sbits,4);
                   if (n>=0) stop_bits=n;
                   else error("stop bits value",arg);
                   break;
         case 'P': n=match(++arg,parit,4);
```

(Continued on page 64)

Add EDITING to your Software with CSE Run-Time™

Your program can include all or a portion of the *C Screen Editor* (CSE).

CSE includes all of the basics of full screen editing plus source in C for only $75. For only $100 more get CSE Run-Time to cover the first 50 copies that you distribute.

Use capabilities like Full cursor control, block move, insert, search/replace or others. Portability is high for OSes, terminals, and source code.

Call for the "CSE Technical Description" and for licensing terms and restrictions.

Full Refund if not satisfied in first 30 days.

Call 800-821-2492

Solution Systems

335-D Washington Street
Norwell, MA 02061
617-659-1571

Circle **no. 75** on reader service card.

FASTER C

TRY FASTER C RISK FREE FOR 30 DAYS WITH OUR MONEY BACK GUARANTEE

helps Develop and Test Lattice C Programs

Reliably Cuts Both —

- Compile times (by 15% to 55%)
- Testing time (by 12% to 37%)

FASTER C keeps the Lattice C library and any other functions you choose in memory. It manages a jump table to replace the LINKER and immediately execute your functions. You can also CALL active functions interactively to speed your program debugging. It includes many options for configuration and control.

AVAILABLE FOR PC-DOS, IBM-AT, AND ANY 256K MSDOS SYSTEM.

ONLY $95.
CALL TOLL FREE
800-821-2492
for "Technical Description" or to order.

FASTER C is a trademark of Solution Systems

Solution Systems™

335-D Washington St., Norwell, Mass. 02061
617-659-1571

Circle **no. 93** on reader service card.

C Helper™

FIRST-AID FOR C PROGRAMS

Save time and frustration when analyzing and manipulating C programs. Use C HELPER's UNIX-like utilities which include:

DIFF and **CMP** – for "intelligent" file comparisons.
XREF – cross references variables by function and line.
C Flow Chart – shows what functions call each other.
C Beautifier – make source more regular and readable.
GREP – search for sophisticated patterns in text.

There are several other utilities that help with converting from one C compiler to another and with printing programs.

C Helper is written in portable C and includes both full source code and executable files for $135 for MS-DOS, IBM AT CPM-80 or CPM-86. Use VISA, Master Card or COD.

Call: **800-821-2492**

Solution Systems™

335-D Washington Street
Norwell, MA 02061
617-659-1571

Circle **no. 94** on reader service card.

PROLOG-86™

Become Familiar in One Evening

Thorough tutorials are designed to help learn the PROLOG language quickly. The interactive PROLOG-86 Interpreter gives immediate feedback. In a few hours you will begin to feel comfortable with it. In a few days you are likely to know enough to modify some of the more sophisticated sample programs.

Sample Programs are Included like:

- **an EXPERT SYSTEM**
- **a NATURAL LANGUAGE INTERFACE** (it generates a dBASE II "DISPLAY" command)
- **a GAME** (it takes less than 1 page of PROLOG-86)

PROTOTYPE Ideas and Applications QUICKLY

1 or 2 pages of PROLOG is often equivalent to 10 or 15 pages in "C" or PASCAL. It is a different way of thinking.

Describe the FACTS and RULES without concern for what the computer will have to do. Maybe you will rewrite in another programming language when you are done.

Programming Experience is not required but a logical mind is. PROLOG-86 supports the de facto STANDARD established in "Programming in Prolog."

AVAILABILITY: PROLOG-86 runs on MSDOS, PCDOS, IBM AT or CPM-86 machines. We provide most formats. The price of PROLOG-86 is **only $125.**

Full Refund if not satisfied during first 30 days.
800-821-2492

Solution Systems™

335-D Washington Street
Norwell, MA 02061
617-659-1571

Circle **no. 95** on reader service card.

```
                    if (n>=Ø) parity=n;
                    else error("parity value",arg);
                    break;
            case 'L': n=match(++arg,wordl,4);
                    if (n>=Ø) word_length=n;
                    else error("word length value",arg);
                    break;
            default : error("switch",arg); break;
        }
    }
    else
        error("switch",arg);
}

match(value,table,size)                 /* Check if switch value is valid */
char *value,                            /* Command line value */
     *table[];                          /* Table of valid arguments */
int size;                               /* No. of entries in table */
{
    int ndx;

    ndx=Ø;
    while ((ndx<size) && (strcmp(value,table[ndx]) != Ø))
        ndx++;
    return( ndx<size ? ndx : -1 );/* Return -1 if not found */
}

get_prm()                               /* Read mode ports to determine */
{                                       /* current port parameters      */
        int mode1,mode2;

        mode1 = input(MODE_REG);/* get mode register 1 */
        mode2 = input(MODE_REG);/* get mode register 2 */

        baud_rate   = getbits(mode2,3,4); /* isolate bit fields */
        stop_bits   = getbits(mode1,7,2);
        word_length = getbits(mode1,3,2);
        parity      = getbits(mode1,5,2);
}

getbits(x,p,n)                          /* get n bits from position p */
unsigned x, p, n;                       /* Ref: K&R pg 45 */
{
        return((x >> (p+1-n)) & ~(~Ø << n));
}

set_prm()                   /* Re-program PCI with new parmameters */
{
        int mode1, mode2;

        mode1 = Øx02 |                      /* set asynch. 16x mode */
                (  stop_bits   << 6) |  /* merge parameters     */
                ( (parity      << 4) & Øx3Ø ) |
                ( (word_length << 2) & ØxØc );

        mode2 = Øx3Ø |                      /* set internal clock   */
                baud_rate;                  /* merge baud rate      */

        output(CMND_REG,DISABLE);           /* disable recv/xmit    */
        output(MODE_REG,mode1);             /* send new parameters  */
        output(MODE_REG,mode2);
        output(CMND_REG,ENABLE);            /* now enable PCI       */
}
```

```
show_parm()                          /* Display current parameters */
{
        printf("\nBaudrate    : %s\n",baudr[baud_rate]);
        printf(  "Stopbits    : %s\n",sbits[stop_bits]);
        printf(  "Parity      : ");
        switch(parity) {
           case Ø: printf("None\n"); break;
           case 1: printf("Odd\n"); break;
           case 2: printf("None\n"); break;
           case 3: printf("Even\n"); break;
        }
        printf(  "Wordlength : %s\n",wordl[word_length]);
}

show_usage()
{
   printf("\nUsage: SETPORT [-Bb][-Ss][-Pp][-Ll]\n\n");
   printf("       where -Bb sets Baudrate,   b = 5Ø..19200\n");
   printf("             -Ss sets Stopbits,   s = Ø, 1, 1.5, 2\n");
   printf("             -Pp sets Parity,     p = O, E, N\n");
   printf("             -Ll sets wordLength, l = 5, 6, 7, 8\n");
}

error(type,value)                    /* Warn command line argument error */
char *type, *value;
{
   printf("Illegal %s ignored : %s\n",type,value);
}

input(port)                          /* Return input from I/O port */
int port;
{
#asm
        pop     b               ;Return address
        pop     h               ;Port number
        push    h               ;Restore stack
        push    b
        mov     a,l             ;Get port into A
        sta     inadr           ;Modify next instruction------+
        in      Ø               ;Dummy input instruction      ¦
inadr   equ     $-1             ;Actual port gets poked here <+
        mov     l,a
        mvi     h,Ø
#endasm
}

output(port,byte)                    /* Output to I/O port */
int port, byte;
{
#asm
        pop     b               ;Return address
        pop     d               ;Byte to output
        pop     h               ;Port number
        push    h               ;Restore stack
        push    d
        push    b
        mov     a,l             ;Get port into A
        sta     outadr          ;Modify next instruction--------+
        mov     a,e             ;Get byte to output             ¦
        out     Ø               ;Dummy output instruction       ¦
outadr  equ     $-1             ;Actual port gets poked here <--+
        mov     l,a
        mvi     h,Ø
#endasm
}
```

End Listing

Christensen Protocols in C

by Donald Krantz

About two years ago, when I acquired my first 8-inch machine, I needed a file transfer program to load data between the 8-inch machine and my 5¼-inch machines. I started looking around for a version of MODEM7 on 8-inch disk.

Before I found one, I ordered a couple of 8088 co-processors for my systems. As this development promised to complicate my file transfer problems considerably, I decided to write a file transfer utility in C so I would have a single, portable program for all my systems.

I started writing in SuperSoft C, because that was what I had back then, and several weeks later had a version working on my Zorba CP/M machine. I thought I'd be in Fat City when the co-processors came.

Well, the co-processors arrived, I wired them in, and I started planning the big port. I was unpleasantly surprised to find that the co-processors had no assembler or documentation and wouldn't read my 5¼-inch IBM disks. As far as I could tell, I had no means of getting any new programs or files into the system short of writing a program with DEBUG. This was unacceptable, especially since I had promised myself that I would never learn nor use 8088 assembly language and had given myself a waiver only to port the source for the file transfer program.

I put the 8088s aside and continued with my life. As time passed, I used the file transfer program, which by now had the name XFR, and gradually started adding bells and whistles.

Some time later, I had occasion to write XMODEM in C for my portable BBS series. I thought I'd just take the old XFR program and do a little surgery; I figured the whole thing ought to take no longer than a couple of hours.

By this time, I had switched all of my new program development to Eco C, which is closer to the Unix Version 7 standard than was SuperSoft. As I started work on XMODEM, I found more and more places where the basic structure of XFR made porting the thing less and less attractive. Still, I took out the digital chain saw and started to work on XFR.

Some of the items I took out include an assembly language CRC generator, interrupt-driven receive FIFO, XDIR-style directory, wildcard filename expansion for the batch mode, file eraser, disk drive logger, half duplex and host terminal mode, and a data line monitor for watching control characters. What's left is a basic Christensen protocol engine with a minimal terminal program. The advantage is that the file transfer primitives are now pretty much independent of the operating system and C compiler—to the extent allowed by the protocol—and can be dropped

The most-often-ported of telecommunications programs meets the most-often-called-portable of languages.

Donald Krantz, 2845 42nd Ave. South, Minneapolis, MN 55406.

into other environments easily.

The basic features supported by XFR are dumb terminal emulation with host system terminal attributes and checksum or CRC file transfer in single or batch file mode. I'll run through the Listing (page 71) and discuss features and portability considerations on a function-by-function basis. First, let me mention up front that my goal was operation at 19.2K-baud, so I have violated some software development standards in the interest of speed, specifically by using C's interfunction GOTO to reduce conditionals in loops. Also, Z80 implementations of C tend to access globals faster than automatics, so globals perhaps are overused.

Function main( )

This is basically a menu processor. Most of the functions do returns rather than using the longjmp( ) to resume the menu loop. It's a trivial job to add features to the menu if you plan on actually using this as a terminal program and need more functions.

Function term( )

This is a simple dumb terminal. For you MSDOS or Unix folks who haven't memorized the CP/M BDOS function calls, the function _bdos(6, 0xFF) is a specialized version of getchar( ) that immediately returns a NULL if no character is waiting. I've seen a similar function, sometimes called inkey( ) or keyscan( ), in other C libraries. The presence or absence of term( ) doesn't affect the file transfers at all, but because you usually should be able to communicate with the remote system to set up the file transfer, it's nice to have some sort of dumb terminal.

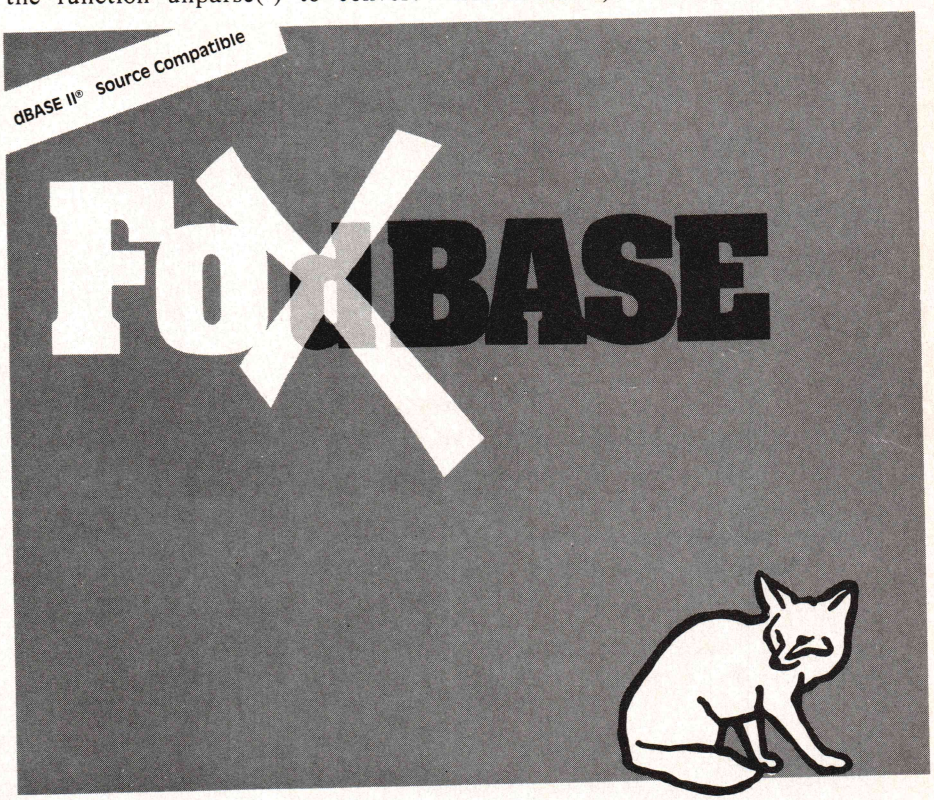
Function transmit( )

This function has the only cheezy user interface in the program. It selects batch file transfer, as opposed to single file transfer, by looking for a comma at the end of the input line when asking for the transmit filename; should it find a comma, it will select batch mode and ask for the next name. Most versions of MODEM7 either have a command option switch for this purpose or recognize that you want batch mode if an ambiguous filename is specified. The original XFR used the latter method. Unfortunately, there is no portable way to expand filenames and get a list of matching files.

If you have a system that will expand wildcards on the command line and pass matching names through the argc/argv method, you could easily convert this program from interactive to command line mode. Otherwise, on CP/M systems, if you have access to the BDOS calls, you can use the function unparse( ) to convert CP/M fcb names to a normal format and pass them one at a time to trans( ). The one advantage the current method has over wildcard expansion is that it allows sending files with dissimilar names. However, if you do use the CP/M "search for file" and "search for next" calls, be sure to collect all the names before you allow the program to proceed.

Function trans( )

This is the file transmission primitive executive (if that's not a contradiction in terms). A filename is passed



Evolution.

Now FoxBASE, the dBASE II source-compatible interpreter/compiler, is even better than before. Automatic 8087 co-processor support allows you to perform numeric computations with lightning speed. Fourteen-digit precision gives you 40% greater accuracy than dBASE II. The fact that FoxBASE is not copy protected means you can easily load it onto your hard disk. What's more, FoxBASE comes complete with a NO-RISK demo plan.

Of course, FoxBASE still offers all of the features that dBASE II does . . . PLUS • Runs 3 to 20 times faster • Permits up to 48 fields/record...50% more than dBASE II • Supports full type-ahead • Compiles program sources into compact object code • Has twice as many variables • Comes with a sophisticated online manual and HELP facility.

FoxBASE is currently available on a wide range of machines: IBM-PC, IBM-PC/XT/AT, COMPAQ & IBM compatibles, TI Professional, DG Desktop, and DG MV-Series to name just a few. And it will soon be available on the Molecular and NCR Tower computers as well. Call or write today for more information.

MS-DOS:	Development Pkg.	$395
	Runtime Pkg.	$695
AOS/VS:	Development Pkg.	$995
	Runtime Pkg.	$1995
UNIX and XENIX:	(To Be Announced)	

Developed by DACOR COMPUTER SYSTEMS

dBASE II is a trademark of Ashton-Tate.
UNIX is a trademark of AT & T.
FoxBASE is a trademark of Fox Software Inc.

FoxBASE™ from FOX SOFTWARE INC.

13330 Bishop Road, P.O. Box 269, Bowling Green, OH 43402 / 419-354-3981 / TWX 810-499-2989

Circle no. 40 on reader service card.

in, in a format compatible with the fopen( ) function. If batch mode is selected, the name is sent to the remote system, followed by the file itself. This function, and the supporting functions it uses, are all pretty much system independent within the confines of the protocol.

Function txrec( )

This function sends a single record to the remote system and does any retransmission necessary due to errors. The record size is fixed at 128 bytes by the macro RECSIZE. If you don't need compatibility with the rest of the world, you could try optimizing performance by increasing the size of the record, if you have a good communication link, or decreasing the size of the record if you use noisy

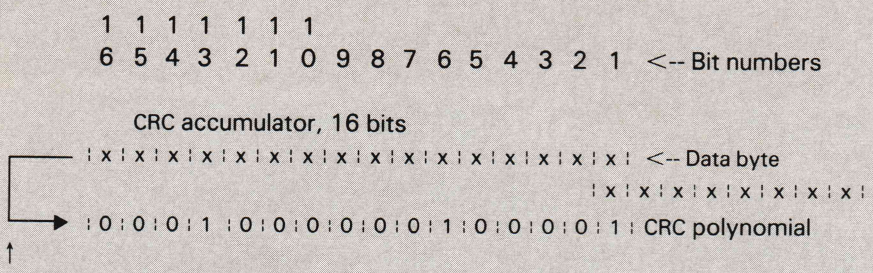


Figure 1.
A 1 shifted out of the left end of the CRC accumulator causes the CRC polynomial to be exclusive-ORed with the post-shift CRC accumulator. The CRC polynomial appears to have no relation to the polynomial equation in the text until you realize that it has been shifted left along with the accumulator.

Byte Number	Definition
1	SOH (Start of Header)*
2	Record number mod 256*
3	[One's complement of record number*
4	Data byte 1
.	
.	
131	Data byte 128
132	CRC high byte†
133	CRC low byte†

* Not included in CRC or checksum calculation
† Included as zeros in CRC calculation

Figure 2.
Christensen packet makeup is shown for a CRC error detection packet. In checksum, byte 133 is not sent, and byte 132 is the checksum.

Programmers writing MODEM7-compatible programs traditionally learn about the protocol by getting as many copies of the source to MODEM7 as possible and tracing program flow. In violation of that tradition, I have written down my interpretation in English. Glock's Fourth Law states that this must have been done before, but the corollary to that law states that you can't find out where, even if it has.

Ward Christensen (according to legend) developed the protocol, which was extended by several other people who ought to receive credit but whose names I don't have. Ward has been around so long that I was surprised, and perhaps a little indignant, to find that he is still alive. He is responsible for some of the more clever public domain programs, so I'm willing to bet that he didn't do some of the peculiar things in the Christensen protocol—such as the filename transfer.

There are two levels to the Christensen (also called XMODEM) protocol: file-at-a-time and batch transfer. The batch transfer mode provides for the unattended transmission of multiple files, including the filename.

The protocol uses two different error detection schemes: checksum and CRC (Cyclic Redundancy Code). Older versions of MODEM7 use the checksum method, while the newer versions use the more accurate CRC method. Most programs that can do CRC will also do checksum.

Only the data bytes in a packet are included in the calculation of the error detection code. The protocol handles errors in headers and handshakes by ignoring garbage characters or assuming the worst.

Checksums are calculated by adding all data bytes mod 256, with any carry ignored.

The CRC polynomial used is the

lines. The macro RECSIZE fixes all the parameters necessary for change of record size.

Function txname( )

This function sends a filename to the remote system when batch mode is selected. The protocol calls for a fixed 11-character field in CP/M fcb format. This part of the protocol has a slightly different flavor to it than the rest, so I suspect that somebody other than Ward Christensen did it. If you're not interested in compatibility with MODEM7, the size of the name is fixed by the macro NAMESIZE. You're safe up to about 30 characters or so before you have to start hunting for other items affected by the name size that NAMESIZE doesn't control.

CCITT $X^{16} + X^{12} + X^5 + 1$. I'm told that there are people who know what that means (I'm taking the word of an old XMODEM listing). In simple language, the procedure for generating the CRC is this: A byte to be added to the CRC is fed into the generator, one bit at a time, high bit first. The bit is shifted into a 16-bit CRC accumulator low end. If the high bit shifted out of the CRC accumulator is a 1, the CRC accumulator is exclusive-ORed with 0x1021. The process repeats for all eight bits of the input character. See Figure 1 (page 68).

In the case of the Christensen protocol, after all the data bytes in a given packet are sent through the CRC generator, two zero bytes are sent through at the end. I'm guessing that these bytes take the place of the CRC bytes in the packet, although I don't know why this is needed. I do know that it doesn't come out right unless you do this, and for an empirical kind of guy like myself that's plenty good enough.

At the single file level of the protocol, both systems are told to begin, and the transmitter waits for the receiver to send a sync byte. The initial sync byte tells the transmitting system whether CRC or checksum will be used. If checksum, the sync byte is a NAK. If CRC, the sync byte is a 'C.' If the receiver is thinking CRC and the transmitter isn't capable of it, both systems hang until they time out. Some programs send five or six 'C' sync bytes and, if not answered, then send NAKs in case the transmitter is an old version.

In CRC mode, once the systems are in sync, the 'C' is dropped in favor of ACK/NAK handshaking for subsequent records.

Once in sync, the transmitter sends a packet consisting of an SOH character, the record number mod 256, the one's complement of the record number, a 128-byte data block, and the error detection byte(s). See Figure 2 (page 68).

The receiver must accept the entire packet at once. Most versions of MODEM7 wait for the SOH, so a missed SOH is an error from which it is difficult to recover. After reception, conditions that will cause record rejection are: the record number doesn't match the one's complement of the record number; the error detection number doesn't match; SOH is not intact; and the record number either is not the record number expected or is one less than the record number expected (indicating a previous ACK was trashed). If the record is rejected, the receiver sends a NAK to the transmitter, which retransmits the record. If the record is accepted, the receiver sends ACKs at about one-second intervals until the next SOH.

This process repeats until all records are transmitted. At the end of file, the transmitter sends EOT at one-second intervals until the receiver ACKs the EOT.

Record numbers are "natural" numbers; that is, the first record is numbered 1, as opposed to the computer natural 0.

Batch level of the protocol includes a filename header in front of the file. Filenames always use checksum error detection, even if CRC mode is selected for the rest of the file. The filename is sent as it would appear in a CP/M file control block (fcb), a fixed block, 11 characters long, with blanks expanding the dot should the filename be less than 11 characters. It is important in CP/M systems to make sure that any high bits are stripped from the filename. See Figure 3 (page 69).

When preparing to receive a name, the receiver sends NAKs at about one-second intervals until the transmitter responds with an ACK followed by the first character of the filename. The receiver then handshakes with ACKs until the transmitter sends EOF. This may mean that more than 11 characters are received (because of line noise). When the receiver gets the EOF, it responds with the checksum of the name, including the EOF byte. If the checksum matches, the transmitter sends an ACK, and the single file process takes over. If the checksum doesn't match, the transmitter sends a NAK, and the process repeats.

If the transmitter sends an EOT in place of any name character, it causes batch transfer mode to terminate. This is the normal end to a batch transfer.

Both the transmitter and the receiver should be able to recognize CAN (Cancel) in place of ACK, NAK, EOT, SOH, or EOF. In practice, timeouts will replace some tests for CAN.

All transmit and receive loops should have timeouts and retries built into them to avoid hanging if the two systems get out of sync.

Standard UART setting for the Christensen protocol is eight bits, no parity, one stop bit.

TEST.DOC becomes:	¦T¦E¦S¦T¦ ¦ ¦ ¦ ¦D¦0¦C¦
VERY.LONG.NAM becomes:	¦V¦E¦R¦Y¦L¦O¦N¦G¦N¦A¦M¦
NAME becomes:	¦N¦A¦M¦E¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦

Figure 3.
CP/M filename expansion

The Best Source Debugger for C

Interactive testing: enter any C expression, statement, or function call, and it is immediately executed and the result displayed. **Direct execution** allows fast and thorough testing, makes learning C a snap.

Run-time checking: execution stops upon exception, and source code displayed. Exceptions include array reference bounds, stack overflow arithmetic or floating point error, etc. Pointers are checked for null or out of range values.

Breakpoints: *any* number of breakpoints can be set *anywhere* in your program; breakpoints are set with screen editor, not by line numbers. Breakpoints may be conditional. Single-step by statement. Interrupt execution from keyboard. Breakpoint, exception, or interruption is always shown with source code. Examine and modify data, look at stack history. Even change your program and then resume execution!

Lint-like **Compile-time checking:** argument number and sizes are checked for consistency. Never mismatch source and object code.

The best feature of all: the *fastest* C interpreter is right there when you're debugging. Make changes in seconds with the integrated screen editor. Test the changes immediately, running your program at compiled speed. Save source code for your favorite compiler, or make stand-alone executable programs. Nothing else is needed. ***Instant-C* is the fastest way to get working, fully debugged C programs available today.**

"We sincerely feel that ***Instant-C*** can have a major positive impact on programmer productivity." *Computer Language*, Feb. 85, pp. 82-83.

Instant-C is only $495. Money back for any reason in first 31 days.

Rational
Systems, Inc.
(617) 653-6194
P.O. Box 480
Natick, MA 07160

Circle **no. 79** on reader service card.

Function receive( )
This is the file reception primitive executive. All it really does is figure out if you want batch mode or single file mode. It's not possible to tell what the other system has selected just by listening to the line. If an asterisk is entered in the receive filename specification, batch mode is selected while preserving any drivespec. Those with non-CP/M systems could remove the drivespec material, but this probably won't be necessary. It's triggered by a colon in the second character position of the filespec.

Because the Christensen protocol is receiver driven, you have to have made a choice whether to use checksum or CRC error detection before you initiate the receive mode. Some versions of MODEM7 will hunt back and forth between CRC and checksum mode to match what the other system uses, but this program doesn't.

Function rxbatch( )
This takes over as receive executive if batch mode is selected. It receives a filename from the remote system and converts it from the fcb format used for transmission back to a format suitable for use with C functions. The name is then passed to the file receive function rxfile( ) to pick up the body of the file.

Function rxname( )
This is the companion to txname( ). Enough said.

Function rxfile( )
This handles reception of an entire file. This function is the time-critical part of the program, which is why I didn't break it into smaller modules. It doesn't matter if the program is slow when transmitting, but when receiving you have to catch all of the characters sent from the other end as they come in. I put off the computation of checksum or CRC to the end of the record for this reason, instead of building it into the receive character function where it probably belongs.

It's also important not to output messages to the display when a packet transfer is in progress. One of the versions of MODEM7.11 that is floating around can't run above 1200 baud in receive mode because it makes a message output call immediately after receiving the packet header and so misses a couple of characters.

Function make( )
This opens a file to receive to. The original version of XFR made a backup file if a file would be overwritten, but file renaming isn't portable in C (the current version will happily destroy an old file, even if reception is terminated prematurely). I left the code in a false conditional assembly block because I thought it rather clever; it's a good feature if your C library supports renaming. The function link takes the arguments (newname, oldname).

Function parse( )
This takes a filespec useable by C functions and converts it to the CP/M fcb format needed by the Christensen protocol. On systems with pathnames, you'll have to hack this function into shape to deal with filename preambles. It will operate on names that don't follow CP/M conventions in most circumstances, although the use of periods will give some unusual results. If the NAMESIZE macro is resized, parse( ) will still operate in a reasonable manner.

Function fillbuf( )
This loads the transmit file buffer with a record. I modularized this section because soon I'll add the ability to extract members from a library file. On systems capable of file sizes that aren't multiples of system records, this function pads the last record with zeros. Note that this code will never be executed on a CP/M system.

Function unparse( )
This undoes what the parse( ) function does. Notice that a period always is inserted into a filename unless the filename is less than (NAMESIZE–3) characters long.

Function updcrc( )
This adds a character to the CRC or checksum accumulator. This uses the CCITT CRC polynomial, but characters are fed into the generator high

bit first. I'm not certain, but I think that makes it incompatible with communications chips that do a hardware CRC, in case you have thoughts in that direction. At any rate, look into it before you spend a lot of time on it.

Function abort( )

This scans the console for an operator-requested abort. As far as I know, this function is called inside any loop where the system can hang (but I never guarantee anything about a program). The BDOS keyboard scan is the same as in term( ).

Function error( )

This is called when fatal transmission or reception errors occur. It ties up loose ends then jumps back to the menu.

Functions wait( ), swait( ), and waitcan( )

These do different delays while waiting for characters. You could combine them all into one function if you're willing to pay the overhead for more conditionals. At high baud rates, file reception timing is pretty critical, but for 1200 baud or so there won't be any problems.

Remaining Functions

In CP/M systems, all of the above functions should work as is. The rest of the functions are all machine dependent—or potentially so.

The init( ) and quit( ) functions are called at the beginning and end of the program. I had code to set up and remove an interrupt-driven FIFO in these functions; UART initialization or I/O channel acquisition would probably go here as well.

The I/O primitives are set up for an I/O-mapped Intel 8351A UART. These functions are the ones most likely to require customization to make the program work.

How does the whole thing perform? On a Z80 running at 3.58 MHz, the program will easily do transfers in checksum mode at 19.2K baud. I'm reasonably sure it will do the same in CRC mode, but I haven't the resources to test CRC at that speed. If anyone does do a benchmark, I'm interested in the result.

As far as error handling, the program is not ready for an end user, but a reasonable person would have to work at it to blow the thing up. Consider this a starting point, and if anyone gets it up on 8-inch MSDOS for the Bigboard and Co-Power 88, let me know.

Source is available on TCOG BBS (612) 724-7779 300/1200 baud.

DDJ

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 196.**

Protocols Listing (Text begins on page 66)

```
/*-----------------------------------------------------------------
                            XFR
                    By Donald G. Krantz 3/84
              (C) Copyright 1984 Donald G. Krantz
                      All Rights Reserved

XFR is a packet protocol file transferer similar to modem7. It is
designed for 19.2K baud transfers "nose-to-nose" between computers
and works very well in modem use at low speeds. XFR is compatible
with most versions of Modem7.

-----------------------------------------------------------------*/

#include "stdio.h"              /* ECO C */

/*-----------------------------------------------------------------
ASCII and XMODEM control characters
-----------------------------------------------------------------*/

#define SOH 1                   /* start of header              */
#define EOT 4                   /* end of transmission          */
#define ACK 6                   /* true acknowledge             */
#define NAK 0x15                /* false acknowledge            */
#define CRC 'C'                 /* request CRC mode             */
#define CAN 0x18                /* cancel transmission          */
#define EoF 26                  /* end of file (used for name)  */
#define BADNAME 0x75            /* received bad name checksum   */
#define BELL 7                  /* console bell                 */
#define NEWLINE 10              /* linefeed char                */
#define CR 13                   /* carriage return char         */
#define BS 8                    /* backspace character          */
#define ESCTERM 'E'-'@'         /* escape from terminal mode    */

/*-----------------------------------------------------------------
User accessable system equates
-----------------------------------------------------------------*/

#define RETRY 5                 /* no. of retrys before abort   */
#define RECSIZE 128             /* transfer record size         */
#define NAMESIZE 11             /* filename fixed length        */
#define TITLE printf( "\n\t\t\t  XFR - File Transfer Utility")
#define VERSION printf("\n\t\t\t ZORBA Version 2.00 as of 3/85")
#define CLS printf( "\033E")    /* clear screen                 */
```

(Continued on next page)

```
#define ERROR -1
#define MAGIC_NUMBER 10000      /* time constant - machine dep. */
#define VOID int                /* function type                */
/* #define LINK                 / * file rename supported       */

/*-----------------------------------------------------------------
Machine dependant port addresses - ZORBA Version (Intel 8251A)
-----------------------------------------------------------------*/

#define RXDATA 0x20             /* rec'd data UART port         */
#define TXDATA 0x20             /* tx'ed data UART port         */
#define UARTCMD 0x21            /* Command/Status port          */


/*-----------------------------------------------------------------
Global variables
-----------------------------------------------------------------*/

char *line;                     /* scratch input line           */
char *buffer;                   /* record buffer                */
FILE *fd;                       /* file POINTER!                */
unsigned rec;                   /* record number                */
char checksum;                  /* checksum accumulator         */
unsigned crcaccum;              /* global crc bytes             */
char crc;                       /* CRC on/off flag              */
char batch;                     /* batch on/off flag            */
jmp_env to_menu;                /* long jump envelope           */
struct {                        /* batch list array             */
                char name_part[ 32 ];
        } list[ 32 ];

/*-----------------------------------------------------------------
Function type declarations
-----------------------------------------------------------------*/

VOID main(), term(), transmit(), trans(), txname(), txrec();
VOID receive(), rxname(), rxfile(), make(), error(), clrcrc();
VOID updcrc(), abort(), sleep(), quit(), init(), tx(), rxbatch();
char *index(), *parse(), *unpa se();
int fillbuf(), wait(), swait(), waitcan(), rxstat(), rx();
int txstat();

/*-----------------------------------------------------------------
Executable code follows.

main() is (of course) the master program loop. Uses setjmp()
to un-nest function calls on error exits.

References globals: crc, tty;
Modifies globals: crc, tty;
-----------------------------------------------------------------*/

VOID main()
{
   char menu;

        menu = FALSE;           /* initially expert mode        */
        init();                 /* initialize modem and UART    */
        TITLE;
        VERSION;
        printf( "\nType 'M' for MENU display....");
        setjmp( &to_menu );     /* menu loop                    */
        while( TRUE )
        {
                printf( "\n\nMode: %s", crc ? "CRC" : "Checksum" );
                if( menu )
                {
                  printf( "\n\n\tS Send file    R Receive file\n" );


                  printf( "\tT FDX term     V CRC/Checksum\n" );
                  printf( "\tM toggle menu  X Exit\n" );
                }
                printf( "\n\nCommand: " );
                switch( toupper( getchar() ) )
                {
                        case 'S':               /* send file(s) */
                                transmit();
                                break;
                        case 'R':               /* RX file(s)   */
                                receive();
                                break;
                        case 'T':               /* dumb terminal*/
                                term();
```

(Continued on page 74)

Periscope ... A Direct Hit!

"A great debugger ... If you've been frustrated by programs that don't behave the same for the debugger as when running alone, or that crash altogether, you should check out Periscope ..."

PC REPORT, Boston Computer Society

"It works, and works well!! In the first day of use I finished up two weeks of problems!! Really excellent!!!"

Peter Loats, Periscope User

Periscope's differences hit home! Its breakout switch gives you control, even when interrupts are disabled. Periscope's unique power helps you solve the difficult problems. Debug device-drivers, memory-resident, and non-DOS programs.

Great for straightening out confused C pointers! Using its breakpoint power, you can stop on reads and writes to ranges of memory. Stop on registers, words, and bytes, too.

It's tough. Periscope's 16K RAM board is write-protected. Runaway programs can't touch crucial debugger code! Examine the system, safely, at any time: Periscope saves the interrupt vectors and buffers it uses, then restores them when done. It uses ROM BIOS calls for functions except file access, so DOS can't clobber itself.

It gets you moving. It's commands are similar to Debug's, so you'll master it quickly. On-line help is there when you need it. You can define the windows you want; you can design easy-to-read memory displays. Periscope supports C, Assembler, BASIC, and Pascal. Comprehensive symbol support gives you a roadmap through memory tying back to your source.

It's risk-free. Periscope is backed up by a 30-day, money-back guarantee! The board is warranted for a year. Technical Support? You get the best there is ... direct from Brett Salter, Periscope's author!

It requires. An IBM PC, XT, AT, Compaq, or close compatible; PC-DOS, 64K RAM; a disk drive and an 80-column monitor. With two monitors, Periscope's screen is displayed on the monitor not in use. With one monitor, a keystroke switches screens.

It's power you can afford. At $295, you can afford Periscope's professional power. The system includes the memory board, breakout switch, debugger software, manual, and quick reference card.

Order now! Call toll-free (800) 722-7006 to place your order or to get more information. MasterCard/VISA accepted.

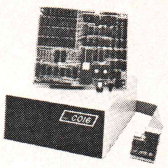
Get your programs up and running;

UP PERISCOPE!

Data Base Decisions/14 Bonnie Lane | Atlanta, GA 30328/(404) 256-3860

Circle **no. 31** on reader service card.

Z80 System Owners Join The 16/32 Bit Revolution Through Evolution



Starting At $695.00

CO-PROCESSING

The most cost effective way for Z80 system owners to obtain 16/32 bit processing power and software compatibility is via the HSC CO-16 Attached Resource Processor.

CO-16 is compatible with any Z80 system running CPM 2.2 or CPM 3. A few examples include:
- KAYPRO 2/4/10 • TRS 2/3/12/16
- AMPRO LITTLE BOARD
- HEATH 89 • SUPERBRAIN
- XEROX 820 • TELEVIDEO 802/803
- MORROW • EPSON QX-10
- LOBO • OSBORNE 1/EXEC
- CROMEMCO • Plus many more

CO-16

Every CO-16 is delivered with • 16/32 bit micro processor • 16 bit Operating System • 256 Kilo RAM • Z80 interface • 16 bit RAM disk driver • CPM80 2.2 RAM disk driver • CPM 2.2 or CPM 3 compatibility • sources with tools • hardware diagrams • board level or case with power supply.

CO-1686

The only Z80 16 bit co-processor includes • INTEL 8086 • 6Mhz no wait states • MSDOS 2.11 • IBM BIOS emulator • Memory expansion to 768K • 8087 math co-processor • 3-channel Real Time Clock • Runs many IBM PC applications • Shares hard disk space with CPM80 • PC diskette compatibility on many systems • CPM86 • Concurrent CPM is coming.

CO-1668

The only Z80 16/32 bit co-processor includes • MOTOROLA 68000 microprocessor • 6 Mhz no wait states • CPM68K • Full "C" compiler with UNIX V7 library and floats • Memory expansion to 1.25 million bytes • NS16081 math co-processor • Real Time Clock • Complete software development environment • 100% file compatible with CPM80 • OS9/68 UNIX look alike coming in February.

Dealer, Distributor and OEM's invited

Hallock Systems Company, Inc.
267 North Main Street
Herkimer, N.Y. 13350
(315) 866-7125

HSC Inc.

Circle **no. 26** on reader service card.

```
                        break;
                case 'V':               /* toggle crc   */
                        crc = !crc;
                        break;
                case 'M':               /* toggle menu  */
                        menu = !menu;
                        break;
                case 'X':               /* quit         */
                        quit();
                default:                /* option bad   */
                        putchar( BELL );
                }
        }
}


/*---------------------------------------------------------------
term() is the dumb terminal section. Escape by typing ESCTERM
character (see ASCII #defines).
Some machine dependent stuff.
---------------------------------------------------------------*/

VOID term()
{
   char ch;                             /* scratch char         */

        CLS;
        printf( "\n\n" );
        while( TRUE )                   /* the loop.            */
        {
                if( rxstat() )          /* if char coming in    */
                        putchar( rx() );/* display local        */
                if( (ch = _bdos( 6, 0xFF )) != NULL )
                        if( ch == ESCTERM )     /* check exit   */
                                return;
                        else
                                tx( ch );       /* disp remote  */
        }
}

/*---------------------------------------------------------------
transmit() is the transmission executive. It requires no input
parameters. This function takes filename(s) as input from the
user, one line at a time. A comma at the end of a filename
signals the intention to enter anther filename on the next
line. More than one filename entered sets "batch" mode. 32
names maximum, for no especially good reason.

Modifies globals: buffer, batch;
---------------------------------------------------------------*/

VOID transmit()
{
   int ct = 0;
   int i;

        printf( "\n" );
        while( TRUE )
        {
                printf( "Transmit File specification: " );
                gets( list[ ct ].name_part );
                if( index( list[ ct ].name_part, ',' ) == NULL )
                        break;
                *index( list[ ct ].name_part, ',' ) = '\0';
                ct++;
        }
        batch = (ct ? TRUE : FALSE);
        for( i=0 ; i <= ct ; i++ )
        {
                while( rxstat() )       /* gobble garbage       */
                        rx();
                trans( list[ i ].name_part );   /* send file    */
        }
        if( batch )
        {
                while( swait() != NAK ) /* await name request   */
                        abort();        /* operator abort scan  */
                tx( ACK );              /* handshake NAK        */
                sleep();                /* decent interval      */
                tx( EOT );              /* end of transmission  */
        }
}
```

(Continued on page 76)

A Proffessional Quality Z80/8080/8085 Disassembler

WHEN YOU NEED SOURCE FOR YOUR CODE
you need REVAS 3

REVAS interactively helps you:

Analyse your software for modification
disassemble files as large as 64K
Assign Real labels in the disassembly
Insert COMMENTS in the disassembly
Generate a Cross Reference (XREF) listing

A 60 page manual shows how the powerful **REVAS** command set gives you instant control over I/O to files, printer, or console; how to do a disassembly; and even how the disassembler works! You get on line help, your choice of assembler mnemonics, control of data interpretation, and calculation in any number base!

REVAS runs in Z80 CPM computers; is available on 8" SSSD (standard), RAINBOW, and other (ask) formats

Price: $90.00 (plus applicable tax), Manual only: $15.00

REVASCO
6032 Chariton Ave., Los Angeles, CA 90056
Voice: (213) 649-3575 *Modem: (213) 670-9465*

Circle **no. 80** on reader service card.

NOW
for Microsoft
C version 3.0

the source debugger for lattice C

Your time and convenience come first! The MSD C Debugger™ is the last, and perhaps final, word in programming assistance for Lattice C users. C Debugger produces a high level view of C programs via function names, line numbers, variable names and C data types, plus a low-level view of machine addresses and instructions for testing assembler language functions.

More features include:

- All documentation is prepared for programmers.
- Online help screen throughout the process.
- Capability to single step through your program.
- Set break points, examine registers and variables.

$165.00 + **$3.50 shipping**

VISA MasterCard

To order, call or write:
MICRO-SOFTWARE DEVELOPERS, INC.
214½ W. Main St. • St. Charles, IL 60174
312/377-5151
Lattice C is a trademark of Lattice, Inc.
Microsoft is a trademark of Microsoft Corp.

Circle **no. 134** on reader service card.

DATESTAMPER™ has the answers

```
Drive B1:   4 files, using 15K (110K FREE)      14:01-09 Feb
  -- file       size      created          accessed         modified --

B1: ADDRES  .DAT   5K | 22:01-17 Jan     08:30-01 Feb     08:23-01 Feb
B1: JSMITH  .LTR   2K | 16:30-24 Dec'84  11:59-10 Feb     16:30-24 Dec'84
B1: TEST1   .BAS   4K | 09:34-22 Jan     16:27-30 Jan     09:35-22 Jan
B1: TEST2   .BAS   4K | 11:55-01 Feb                      11:55-01 Feb
```

When did we print that letter?

Has the mailing list been updated?

Which is the latest version?

DateStamper™ keeps your CP/M computer up-to-date!

- avoid erasing the wrong file
- back-up files by date and time
- keep dated tax records of computer use
- simplify disk housekeeping chores

OPERATION: DateStamper extends CP/M 2.2 to automatically record the date and time a file is created, read or modified. DateStamper reads the exact time from the real-time clock, if you have one; otherwise, it records the order in which you use files. Disks initialized for datestamping are fully compatible with standard CP/M.
INSTALLATION: Default (relative-clock) mode is automatic. Configurable for any real-time clock, with pre-assembled code supplied for popular models. Loads automatically at power-on. **UTILITIES:** • Enhanced SuperDirectory • Powerful, all-function DATSWEEP file-management program with date and time tagging • Disk-initializer • Installation and configuration utilities **PERFORMANCE:** Automatic. Efficient. Invisible. Compatible.

Requires CP/M 2.2. Uses less than 1K memory. Real-time clock is optional.

When ordering please specify format

8" SSSD, Kaypro, Osborne Formats **$49**
For other formats (sorry, no 96 TPI) add $5.
Shipping and handling **$3**
California residents add **6%** sales tax

MasterCard and Visa accepted

Specialized versions of this and other software available for the Kaypro.
CP/M is a registered trademark of Digital Research, Inc.

Write or call for further information

Plu*Perfect Systems

BOX 1494 • IDYLLWILD, CA 92349 • 714-659-4432

Circle **no. 69** on reader service card.

TURBO Power

PROGRAMMER'S UTILITIES
especially for Turbo Pascal on
IBM PC/XT/AT and compatibles

MORE POWERFUL THAN UNIX UTILITIES!!!

Whether you are a
-- Student -- Hobbyist -- Professional Software Developer --
THESE UTILITIES WILL IMPROVE YOUR
PROGRAMMING PRODUCTIVITY!!!

These Powerful, Ready-to-Use programs fully support Turbo Pascal versions 2.0 and 3.0, and MSDOS 2.X and 3.0. Here's what you get:

Pretty Printer
Standardize capitalization, indentation, and spacing of source code. Don't waste your own time! Several adjustable parameters to suit your tastes (works with any standard Pascal source).

Program Structure Analyzer
Find subtle problems the compiler doesn't: uninitialized and unused variables, modified value parameters, "sneaky" variable modification, redefined standard identifiers. Also generates a complete variable cross reference and a program hierarchy diagram. Interactive or write to file (works with any standard Pascal source).

Execution Timer
Obtain a summary of time spent in each procedure and function of your program, accurate to within 200 microseconds. Also counts number of calls to each subprogram. Fully automatic.

Execution Profiler
Obtain a graphic profile of where your program spends its time. Interactive, easy-to-use. Identify weak code at the instruction level. (Profiler and Timer for Turbo Pascal Source code only.)

Command Repeater
Go beyond MSDOS batch files to combine a powerful text parser with general-purpose command execution capability. Use to copy, print or delete across subdirectories, "make" programs and more.

Pattern Replacer
Find and REPLACE versatile regular expression patterns in any text file. Supports nesting, alternation, tagged words and more. Over a dozen programmer's applications included.

Difference Finder
Find differences between two text files, and optionally create an EDLIN script which rebuilds one from the other. Disregard white space, case, arbitrary characters and Pascal comments if desired.

Super Directory
Replace PCDOS DIR command with extended pattern matching, sort capability, hidden file display, date filtering, and more.

File Finder
Locate files anywhere in the subdirectory tree and access them with a single keystroke. Display the subdirectory tree graphically.

AVAILABLE IN SOURCE AND EXECUTABLE FORMAT

Executable: $55 COMPLETE including tax and shipping. Compiled and ready to run, includes user manual, reference card and one 5¼" DSDD disk. Ideal for programmers not using Turbo.

Source: $95 COMPLETE including tax and shipping. Includes all of the above, and two additional DSDD disks. Disks include complete Turbo Pascal source code, detailed programmer's manual (on disk) and several bonus utilities. Requires Turbo Pascal 2.0 or 3.0.

Requirements: MSDOS 2.X or 3.0, 192K RAM — programs run in less RAM with reduced capacity. Two drives or hard disk recommended.

TO ORDER:
VISA/MasterCard orders, call 7 days toll-free 1-800-538-8157 x830. In California, call 1-800-672-3470 x830 any day.
Or mail check/money order to:

TurboPower Software
478 W. Hamilton Ave., Suite 196
Campbell, CA 95008

Circle **no. 81** on reader service card.

Protocols Listing

(Listing continued, text begins on page 66)

```
/*-------------------------------------------------------------
trans() does transmission of a file.

References globals: buffer, fd, batch;
Modifies globals: fd, rec, crc;
-------------------------------------------------------------*/

VOID trans( in_name )
   char *in_name;
{
   char name[ NAMESIZE+1];       /* expanded name                */

   char ch;                      /* scratch character            */
   register char *i;             /* buffer pointer (index)       */
   int y;                        /* scratch counter              */

        rec = 1;                 /* uses "natural" numbering     */
        if( (fd = fopen( in_name, "rb" )) == NULL)
        {
                printf( "Unable to open: %s\n", in_name );
                return;
        }
        if( batch )              /* need to send name?           */
                txname( parse( in_name, name ) );
        printf( "\nTransmitting: %s", in_name );
        while( rxstat() )
                rx();
        while( TRUE )            /* handshake w/ receiver        */
        {
                printf("\nSyncronizing: " );
                if( (ch = waitcan( 10 )) == NAK )
                {
                        printf( "Received Checksum Request\n" );
                        crc = FALSE;
                        break;
                }
                if( ch == CRC )
                {
                        printf( "Received CRC Request\n" );
                        crc = TRUE;
                        break;
                }
        }
        while( fillbuf() )               /* file not empty?      */
                txrec( buffer );         /* send record          */
        tx( EOT );
        while( (ch = waitcan( 1 )) != ACK )
        {
                tx( EOT );               /* show end             */
        }
        fclose( fd );                    /* dump file            */
        printf( "\n%s transferred.", in_name );
}


/*-------------------------------------------------------------
txname() transmits a file name from the CP/M fcb format
parameter 'name'.

References globals: buffer, checksum, crc;
Modifies globals: crc;
-------------------------------------------------------------*/

VOID txname( name )
   char *name;
{
   register int i;               /* scratch counter              */

   char ch;                      /* scratch char                 */
   char crcsav;                  /* holds state of global 'crc'  */

        crcsav = crc;            /* we always use checksum for   */
        crc = FALSE;             /* name error detection         */
        while( TRUE )            /* main loop for retries        */
        {
                while( rxstat() )
                        rx();
                i = 0;           /* retry count                  */
                while( TRUE )
                {
                        printf( "\nWaiting for filename request");
                        if( waitcan( RETRY ) == NAK )
                                break;
                        if( i++ ) RETRY )
```

```
                        error( "Can't send filename" );
                    printf( " - Received garbage" );
                }
                tx( ACK );          /* handshake name request        */
                sleep();            /* decent interval               */
                printf( "\nSending name: " );
                clrcrc();           /* clear checksum accumulator    */
                for( i = 0 ; i < NAMESIZE ; i++ )/* name loop        */
                {
                        tx( name[ i ] );
                        putchar( name[ i ] );   /* local echo        */
                        wait( 10 );     /* wait for ACK              */
                }
                tx( EoF );              /* name terminated w/EoF*/
                if( wait( 10 ) == checksum )    /* handshake     */
                {
                        printf( " - Name sent OK" );
                        tx( ACK );      /* handshake OK checksum*/
                        crc = crcsav;   /* replacce 'crc'        */
                        return;         /* normal exit           */
                }
                tx( BADNAME );          /* handshake bad chksm   */
                printf( " - Bad name transfer" );
        }
}


/*-----------------------------------------------------------------
txrec() transmits a single record, with retransmit on error from
receiver. Input is a pointer to the I/O buffer.

References globals: rec, crcaccum, checsum, crc;
Modifies globals: rec;
-----------------------------------------------------------------*/

VOID txrec( buf )
   char *buf;
{
   register int i;
   unsigned cr;

        while( TRUE )                   /* do it until right     */
        {
                printf( "\rTransmitting record %d   ", rec );
                tx( SOH );              /* start of header       */
                tx( rec );              /* rec #                 */
                tx( ~rec );             /* 1's comp              */
                clrcrc();               /* clear CRC accum       */
                for( i = 0 ; i < RECSIZE ; i++ )
                        tx( buf[ i ] ); /* send record           */
                updcrc( 0 );            /* finish up CRC         */
                updcrc( 0 );            /* again                 */
                cr = crcaccum;          /* save crc lobyte       */
                if( crc )               /* send hi byte first    */
                {
                        tx( crcaccum >> 8 );
                        tx( cr );
                }
                else
                        tx( checksum );
                if( waitcan(10)==ACK )  /* quit if correct       */
                        break;
                printf( "Error in transmission\n" );
        }
        rec++;                          /* bump record count     */
}


/*-----------------------------------------------------------------
receive() is the receive executive. It requires no input
parameters except 'crc' should be set. Does not "hunt" from
CRC to checksum.

References globals: line;
Modifies globals: line;
-----------------------------------------------------------------*/

VOID receive()
{
        printf( "\n\nReceive File specification: " );
        gets( line );
        if( rxstat() )
                rx();                   /* gobble trash          */
        if( index( line, '*' ) != NULL )/* 'batch' set by a      */
                rxbatch( line );        /* asterisk              */
                                        /* in input file name    */
        else
                rxfile( line );         /* otherwise nobatch     */
}
```

(Continued on next page)

REMOVE

from your **C** programs with

PC–LINT

PC–LINT analyzes your C programs (one or many modules) and uncovers glitches, bugs, quirks and inconsistencies. It will catch subtle errors before they catch you.

PC–LINT resembles the Lint that runs on the UNIX O.S. but with more features and greater sensitivity to the problems of the 8086 environment.

- Full K&R C
- Supports Multiple Modules—finds inconsistencies between declarations and use of functions and data across a set of modules comprising a program.
- Compares function arguments with the associated parameters and complains if there is a mismatch or too many or too few arguments.
- All warning and information messages may be turned on and off globally or locally (via command line and comments) so that messages can be tailored to your programming style.
- All command line information can be furnished indirectly via file(s) to automate testing.
- Use it to check existing programs, programs about to be exported or imported, as a preliminary to compilation, or *prior* to scaling up to a larger memory model.
- All one pass with an integrated preprocessor so it's very fast.
- Has numerous flags to support a wide variety of C's, memory models, and programming styles.
- **Introductory Price: $98.00 MC, VISA**
  (Includes shipping and handling) PA residents add 6% sales tax. Outside USA add $10.00.
- Runs on the IBM PC (or XT, AT or compatible) under DOS 2.0 and up, with a minimum of 128KB of memory. It will use all the memory available.

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426

(215) 584-4261

*Trademarks: IBM *(IBM Corp.)*, PC–LINT *(Gimpel Software)*, UNIX *(AT&T)*

Mac Inker

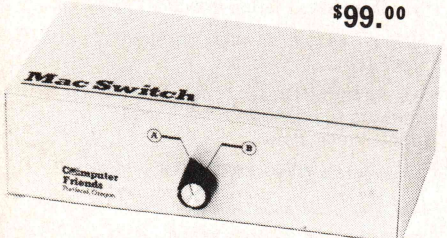
Re-ink any fabric ribbon AUTOMATICALLY for less than 5¢. Extremely simple operation with built-in electric motor. We have a MAC INKER for any printer: cartridge/spool/harmonica/zip pack. Lubricant ink safe for dot matrix printheads. Multicolored inks, uninked cartridges available. Ask for brochure. Thousands of satisfied customers.

$54.95 +

Mac Switch

Mac Switch lets you share your computer with any two peripherals (serial or parallel). Ideal for word processors—never type an address twice. Ask us for brochure with tips on how to share two peripherals (or two computers) with MAC SWITCH. Total satisfaction or full refund.

$99.00



Order toll free 1-800-547-3303

Dealer inquiries welcome

Computer Friends

6415 SW Canyon Court #10
Portland, Oregon 97221
(503) 297-2321

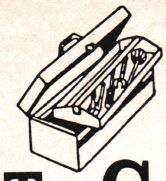
Circle **no. 25** on reader service card.

Protocols Listing

(Listing continued, text begins on page 66)

```
/*-----------------------------------------------------------------
rxbatch() is the batch mode reception executive.

References Globals: line;
Modifies Globals: line;
-----------------------------------------------------------------*/

VOID rxbatch()
{
    char fcb[ NAMESIZE + 1 ];     /* used by unparse()            */
    char name[ NAMESIZE + 5];     /* unparsed name                */

        while( TRUE )
        {
                while( rxstat() )         /* gobble trash         */
                        rx();
                printf( "\nWaiting for file name:  " );
                rxname( fcb );  /* receive file name into fcb     */
                unparse( name, fcb );    /* make "standard"       */
                if( line[ 1 ] == ':' )
                        line[ 2 ] = '\0';
                else
                        line[ 0 ] = '\0';
                strcat( line, name );
                rxfile( line );           /* receive file         */
        }
}


/*-----------------------------------------------------------------
rxname() loads a CP/M style fcb with a filename from remote
sender. On receipt of an EOT instead of a name character,
aborts to master menu via longjmp().
-----------------------------------------------------------------*/

VOID rxname( fcb )
   char *fcb;                     /* points to CP/M style fcb     */
{
   char *fcbptr;                  /* index to fcb                 */
   register int i;                /* scratch counter              */
   int ch;                        /* scratch char (must hold ERR) */
   char chksum;                   /* checksum accumulator         */

        while( TRUE )
        {
                fcbptr = fcb;             /* align index          */
                i = RETRY * 5;            /* retry for name h.s.  */
                while( TRUE )             /* Handshake NAK        */
                {
                        abort();          /* check operator abort */
                        tx( NAK );
                        if( swait() == ACK )
                                break;
                        if( !( i-- ) )
                                error( "Timed out waiting for name" );
                }
                chksum = EoF;             /* init checksum        */
                for( i = 0 ; i < 34 ; i++ )     /* accept noise   */
                {
                        if( (ch = swait()) == EOT )
                        {
                                tx( ACK );
                                printf(
                                "\nAll transfers completed" );
                                longjmp( &to_menu, 0 );
                        }
                        if( ch == EoF ) /* End of name chars      */
                                break;
                        if( ch != ERROR )         /* not timeout  */
                        {
                                *(fcbptr++) = ch & 0x7F;
                                putchar( ch );
                                chksum += ch & 0x7F;
                        }
                        abort();          /* operator abort       */
                        tx( ACK );        /* handshake name char  */
                }
                fcb[ NAMESIZE ] = 0;      /* terminate name field */
                do {
                        abort();          /* operator abort       */
                        tx( chksum );     /* handshake checksum   */
                } while( (ch = swait()) == ERROR );
```

(Continued on page 80)



The Tools You Need To C You Thru.

Now the *WizardWare*™ Applications Programmers Toolkit provides everything you need to increase your C programming productivity.

APT™ features include:

- COMPLETE SOURCE CODE (over 5000 lines!)
- File handling with direct & keyed access
- Screen and Report Generators, with full screen handling for your programs
- Generic Terminal Driver for portable code
- String math functions, and string manipulation routines
- Reference Manual on Disk (over 50 pages)
- Tutorial Manual (over 25 pages) with Source for Mailing List Manager
- A host of useful Utilities, Database and File Editors
- Available for Lattice C, Mark Williams C, DeSmet C, BDS C, others.

Also Available: C-STARTER Toolkit, great for learning C!! Includes: Customized APT, DeSmet C Compiler, and "Programming in C on the IBM-PC" (200 pages)

APT/MS-DOS versions	$495
APT/DeSmet C version	$395
APT/BDS C version	$395
C-Starter (binary APT, DeSmet Compiler and Book)	$295
APT/Manual only	$ 50

Detailed Brochures on request

*Manual Cost will be applied if APT purchased within 30 days ($10 re-stocking charge.) U.S. funds only, please.

Trademarks: MS-DOS/Microsoft, Lattice C/Lattice, Inc., MWC86/Mark Williams Co., DeSmet C/C Ware, CI-C86/Computer Innovations, Inc., BDS C/BD Software, DR-C/Digital Research, WizardWare, APT, C-Starter/Shaw☆American Technologies.

Call (502) 583-5527

Ask for APT™ or C-Starter, or Send Check to:

Shaw☆American Technologies

WizardWare™

830 South Second St. - Box 648
Louisville, KY 40201, USA

(C.O.D. and Foreign Orders - Add $5 Shipping/Handling)

References: Bank of Louisville, Citizens Fidelity Bank, Louisville Chamber of Commerce

Circle **no. 86** on reader service card.

Poor Person Software

Introduces

Write-Hand-Man

Desk accessories for CP/M

Write-Hand-Man lets you take notes, check phone numbers, make appointments, and countless other tasks without leaving Wordstar, dBase, Multiplan, or any other application. Enter **Write-Hand-Man** with a single keystroke and choose the program you want. When you leave **Write-Hand-Man,** your application continues normally.

$49.95 plus tax delivers **Write-Hand-Man** and 4 companion programs; **Notepad, Phonebook, Calendar,** and **Termcomm.** User written programs are easily added. All you need is M80 or some other LINK-80 compatible assembler.

Other CP/M products available from **Poor Person Software: Poor Person's Spooler** ($49.95), **Poor Person's Spelling Checker** ($29.95), **Poor Person's Spread Sheet** ($29.95), **Keyed Sequential Files** ($39.95), **Poor Person's Menus** ($29.95), **aMAZEing Game** ($29.95), **Window System** ($29.95), **Crossword Game** ($39.95), **Mailing Label Processor** ($29.95). Shipping included.

All products available on IBM 8 inch and Northstar 5 inch disks. Other 5 inch formats add $5 handling charge. No credit cards.

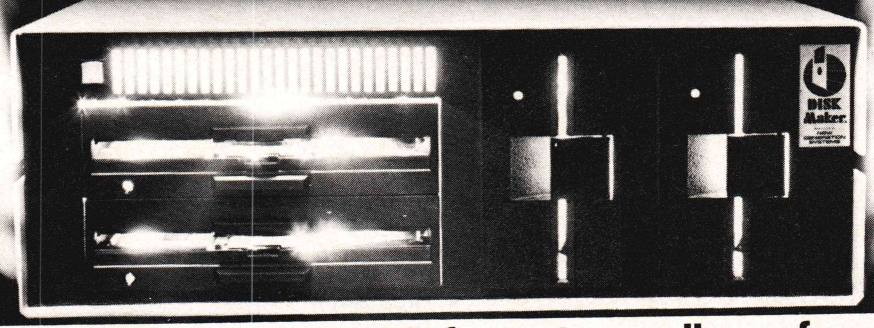
Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
tel 415-493-3735

CP/M is a registered trademark of Digital Research

Circle **no. 71** on reader service card.

The Most Valuable Disk Maker in the Universe

NEW: Disk Maker II Plus



Download fast, read over 200 formats easily, reformat rapidly

The more disk formats you work with, the more our Disk Maker™ system saves time and money by reading and/or writing disks in any of over 200 formats. No modems, no patches, no other special software necessary.

Disk Maker II Plus is a complete, stand-alone system featuring both 8'' and 5¼'' disk drives with options for 3½'' and IBM PCAT drives—up to 12 total drives. With its 6 MHz processor and RAM memory drive, Disk Maker II Plus is fast! Hardware options include a 15 or 20 Mb hard disk, an 8086 MSDOS coprocessor and memory expansion.

Optional software includes word processor and Compugraphic typesetter formats and the MicroDrive Tester for drive alignment testing.

Supported with comprehensive manual, 6 month warranty and continuing software update program.

Disk Maker I is an alternative for existing S-100 systems (CompuPro, Zenith Z-100, North Star, etc.) to provide the same functions as Disk Maker II Plus.

Write or call today for information on how Disk Maker™ can save you time and money.

Jerry Pournelle wrote about Disk Maker™:
''It's easy to install, easy to use, and darned near unique. If you need one, you need it bad.'' (BYTE, July 1984)

NEW GENERATION SYSTEMS

1800 Michael Faraday Drive. Suite 206. Reston. VA 22090
(703) 471-5598 Order Line: (800) 368-3359
Dealer inquiries welcomed.

Circle **no. 76** on reader service card.

```
            if( ch == ACK )          /* ACK is good name      */
                    return;
            printf(" - Checksum error\nRetrying file name: " );
        }
}


/*---------------------------------------------------------------
rxfile() receives a file. Input parameter is the name, including
drivspec if any, to save incoming data under. To maintain speed
the function is rather long to reduce function calls where
possible. Have 'crc' set before entry.

References globals: buffer, fd, rec, crcaccum, checksum, crc;
Modifies globals: buffer, rec;
---------------------------------------------------------------*/

VOID rxfile( name )
   char *name;
{
   char ch;                      /* scratch handshake var        */
   char response;                /* ACK/NAK/CRC handshake        */
   char crcrlo;                  /* rec'd CRC low byte           */
   char crcrhi;                  /* rec'd CRC hi byte            */
   char r1;                      /* current record number        */
   char r2;                      /* 1's comp record number       */
   unsigned int j;               /* wait loop timer              */
   int i;                        /* scratch counter              */
   register char *bptr;          /* buffer index                 */

        while( rxstat() )
                rx();

        make( name );            /* open/make backup             */
        printf( "\n" );
        rec = 1;                 /* uses natural numbering       */
        if( crc )                /* set initial handshake to     */
                response = CRC; /* CRC or checksum               */
        else
                response = NAK;
        while( TRUE )            /* record receive loop          */
        {
                bptr = buffer;           /* align index          */
                printf( "\rWaiting for record %d       ", rec );
                for( i=1 ; i <= RETRY * 5 ; i++ )
                {
                        abort();
                        tx( response ); /* send handshake        */
                        if( (ch = swait()) == SOH )
                                break;  /* SOH indicatees rec    */
                        if( ch == EOT ) /* EOT indicates done    */
                        {
                                printf( "\n%s received OK", name );
                                fclose( fd );
                                tx( ACK );       /* handshake    */
                                return;          /* normal exit  */
                        }
                        if( ch == CAN ) /* Xmit request abort    */
                        {
                                fclose( fd );
                                error("\rReceived cancel request");
                        }
                        if( i == RETRY * 5 )     /* timeout exit */
                                error( "Can't sync to sender" );
                }
                r1 = swait();            /* record number        */
                r2 = swait();            /* 1's comp record #    */
                while( bptr - buffer < RECSIZE ) /* test count   */
                {
                        *(bptr++) = swait();     /* accept char  */
                }
                if( crc )                /* get hibyte CRC       */
                        crcrhi = swait();
                crcrlo = swait();        /* lobyte CRC or chksm  */
                response = NAK;          /* init response        */
                if( (~r1 & 0xFF) != (r2 & 0xFF) )
                {
                        printf( "- Record numbers don't match\n" );
                        continue;
                }
                clrcrc();                /* calc checksum/CRC    */
                for( j = 0 ; j < RECSIZE ; j++ )
                        updcrc( buffer[ j ] );
```

(Continued on page 82)

DE SMET C

8086/8088 Development Package

$109

FULL DEVELOPMENT PACKAGE

- Full K&R C Compiler
- Assembler, Linker & Librarian
- Full Screen Editor
- Execution Profiler
- Complete STDIO Library (>120 Func)

Automatic DOS 1.X/2.X SUPPORT

BOTH 8087 & S/W FLOATING POINT

OVERLAYS

OUTSTANDING PERFORMANCE

- First and Second in AUG '83 BYTE benchmarks

SYMBOLIC DEBUGGER $50

- Examine & change variables by name using C expressions
- Flip between debug and display screen
- Display C source during execution
- Set multiple breakpoints by function or line number

DOS LINK SUPPORT $35

- Converts DeSmet.O to DOS.OBJ Format
- LINKs with DOS ASM
- Uses Lattice® naming conventions

CWARE CORPORATION

P.O. Box C, Sunnyvale, CA 94087

(408) 720-9696

Street Address: 505 W. Olive, #767 (94086) Call for hrs.

All orders shipped UPS surface on IBM format disks. Shipping included in price. California residents add sales tax. Canada shipping add $5, elsewhere add $15. Checks must be on U.S. Bank and in U.S. Dollars. Call 9am–1pm to CHARGE by VISA/MC/AMEX.

Foreign Distributors: **AFRICA,** HI-TECH SVCS, Gaborone 4540 or Telex 2205BD LANGER • **ENGLAND:** MLH Tech, 0606-891146 • **JAPAN:** JSE 03-486-7151 • **SWEDEN:** ESCORT DATA 08-87 41 48 **or** THESEUS KONSULT 08-23 61 60

Circle **no. 18** on reader service card.

WIZARD C

"...written by someone who has been in the business a while. This especially shows in the documentation."

Computer Language
February, 1985

- All UNIX System V language features
- Support for 8087, 80186 and 80286
- Full library source code included
- Cross-file checks (full UNIX lint)
- Uses MS-LINK or PLINK 86
- ROMable data options
- In-line assembly language
- Cross compilers available
- Third party software available, including PANEL

The new standard for C Compilers on MSDOS!

Only $450.

(617) 641-2379

WIZARD
Systems Software, Inc.

11 Willow Court
Arlington, MA 02174

VISA MasterCard

Circle **no. 116** on reader service card.

2 Megabyte SemiDisk!

Have you been waiting on your slow floppy disk drives too long? SemiDisk Systems has a disk emulator for you! It'll put you in the fast lane, with ultra-fast data transfer, huge storage capacity, convenient battery backup, and a handy print spooler.

Have you been waiting for a SemiDisk big enough to handle your large applications programs, files, and databases - all at once? Your wait is over. SemiDisk Systems is now delivering 2 megabytes of disk storage on a single board!

512k, 1Meg and 2Megabyte SemiDisks are available for S-100 computers, (including the H/Z-100 operating under Z-DOS), IBM PC, XT, & AT, the TRS-80 Models II, 12, & 16, and the Epson QX-10. Once you've tried a SemiDisk you'll know why we say. . .

Someday you'll get a SemiDisk.
Until then, you'll just have to . . . . . .wait.

	512K	1Mbyte	2Mbyte
SemiDisk I, S-100	$995	$1795	
SemiDisk II, S-100	$1295	$2095	$2549
IBM PC, XT, AT	$945	$1795	$2499
QX-10,QX-16	$799		$2499
TRS-80 II,12,16	$995	$1795	$2499
Battery Backup Unit	$150		

SEMIDISK

SemiDisk Systems, Inc.
P.O. Box GG, Beaverton, Oregon 97075
503-642-3100

VISA master charge

Call 503-646-5510 for CBBS/NW, 503-775-4838 for CBBS/PCS, and 503-649-8327 for CBBS/Aloha, all SemiDisk-equipped computer bulletin boards (300/1200 baud). SemiDisk, SemiSpool trademarks of SemiDisk Systems.

Circle **no. 85** on reader service card.

```
                updcrc( 0 );                    /* required to finish   */
                updcrc( 0 );                    /* off CRC - why?       */
                if( crc )                       /* CRC test             */
                        if( (crcrlo + (crcrhi (( 8)) != crcaccum )
                        {

                                printf( "- Bad CRC received\n" );
                                continue;
                        }
                if( !crc )                      /* checksum test        */
                        if( crcrlo != checksum )
                        {
                                printf("- Bad checksum received\n");
                                continue;
                        }
                if( (rl == (rec - 1) & 0xFF ) ) /* duplicate?           */
                {
                        printf( " - Received duplicate record\n" );
                        response = ACK; /* dup is OK - ACK was  */
                        continue;       /* trashed - ignore it  */
                }
                if( rl != (rec & 0xFF) )/* fatal sequence error */
                        error( "File record numbering error" );
                rec++;                          /* bump record count    */
                for( j = 0 ; j ( RECSIZE ; j++ ) /* write data  */
                        putc( buffer[ j ], fd );
                response = ACK;                 /* normal loop end      */
        }
}


/*-----------------------------------------------------------------
make() opens a file for output. It renames any existing file
with the same name to ".BAK", and erases any previous backup.
Input parameter is filespec in "normal" (compressed) format.
Note: file rename by the link() function isn't supported
by several CP/M C Compilers, and others call this function
rename(), or reverse the arguments. At any rate, you can determine
how to handle backups in your own system and take any
appropriate action.

References globals: fd;
Modifies globals: fd;
-----------------------------------------------------------------*/

VOID make( name )
   char *name;
{
#ifdef LINK
   char bak[ NAMESIZE + 5 ];    /* holds .BAK filespec          */

                                /* does this name exist?        */
        if( (fd = fopen( name, "r" )) != NULL )
        {
                fclose( fd );   /* we were just checking        */
                strcpy( bak, name );    /* save old name        */
                if( index( name, '.' ) )/* extension spec'ed?   */
                        strcpy( index( bak, '.' ), ".BAK" );
                else
                        strcat( bak, ".BAK" );
                unlink( bak );  /* dump any old .BAK file       */

                link( bak, name ); /* rename new .BAK file      */
        }
#endif
        if( (fd = fopen( name, "wb" )) == NULL )
                error( "Can't create file - Aborting" );
}


/*-----------------------------------------------------------------
parse() expands  non - ambiguous filespecs to the
standard CP/M fcb format, excluding drive byte.
Sets the globals
'batch' if ambigous, and 'driv' for drivespecs.
Inputs are "normal" filespec (inspec), and exanded fcb.
-----------------------------------------------------------------*/

char *parse( inspec, fcb )
   char *inspec, *fcb;
{
   register int i;                      /* fcb index            */
   int inptr;                           /* input spec index     */
```

(Continued on page 84)

Changing Your Address?

Staple your label here.

To change your address, attach your address label from the cover of the magazine to this coupon and indicate your new address below.

Name

Address

Address Apt. #

City State

Zip

Mail to: Dr. Dobb's Journal, PO Box 27809, CA 92128

Lattice presents

LMK

Automated Product Generation Utility

The new LMK Automated Product Generation Utility (similar to UNIX "MAKE") for MS-DOS makes rebuilding your complex software systems or documents a much "lighter" task.

All you do is create "dependency files" consisting of macro definitions and dependency descriptions followed by executable commands to specify relationships of the various pieces of your system. After any changes, a single LMK command initiates the automatic rebuilding of your system.

LMK reads the dependency files and compares the dates and times of the last modification. If the date and time of the target is earlier than the date and time of the area it depends upon, the "action" specified in the file will be executed.

LMK actions can be any executable command. Invoke a batch text editor to make replacements in a number of files. Apply a file comparator to new and old files. Update a database. Or run any other utility. You are limited only by your imagination.

Lighten your programming, maintenance & filing . . . Order LMK today!

PRICE: $195
ASK ABOUT OUR "TRADE UP TO LATTICE C POLICY"

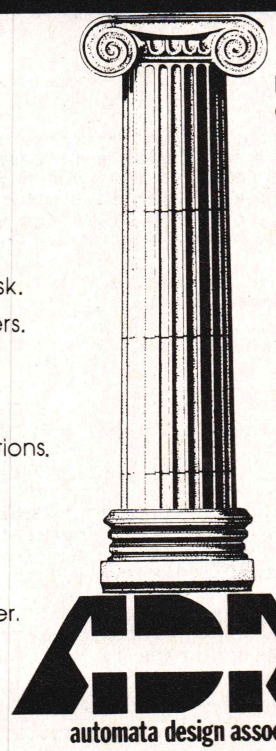
For more information or to place your order contact:

LATTICE®
P.O. Box 3148
Glen Ellyn, IL 60138
TWX 910-291-2190 (312) 858-7950

Circle **no. 58** on reader service card.

THE SYMBOLIC *SINE QUA NON*
L I S P

THE fifth generation language
P R O L O G



automata design assoc.

Cybermetrics **UNXLISP-86** (tm) Features:

- 1 Megabyte address space.
- LAMBDA, NLAMBDA, MACRO with displacement.
- CATCH, THROW, ERRSET, & ERR.
- SAVE/RESTORE virtual images to/from disk.
- &OPTIONAL, &REST, and &AUX parameters.
- Tree-structured object list, functional directories.
- Lisp utilities with source, including pretty-printer, structure editor, debugging functions.
- Works with any text editor through exec function.
- Extensive manual – written in ENGLISH
- Requires 320K memory and runs under MSDOS and PCDOS versions 2.0 and later.

TECHNICAL
(408) 725-1344

$49.00

Implementing the full Edinburgh Syntax as described by Clocksin and Mellish.

Recognized by Japan as providing unparalleled opportunity for artificial intelligence.

Applications:

- The highest level of a hierarchial robotic control system.
- Machine recognition of natural language.
- Expert systems and knowledge engineering.

Optional:

- Virtual memory
- Special libraries
- Language extensions
- Large model

Requires 192K memory and runs under MSDOS and PCDOS versions 2.0 and later.

TECHNICAL
(215) 646-4894

educational package
$29.95
other versions $50–$500

1570 ARRAN WAY
DRESHER, PA 19025

ADD $3.00 FOR SHIPPING & HANDLING

TO ORDER CALL **(215) 355-5400** USE IT FOR 30 DAYS

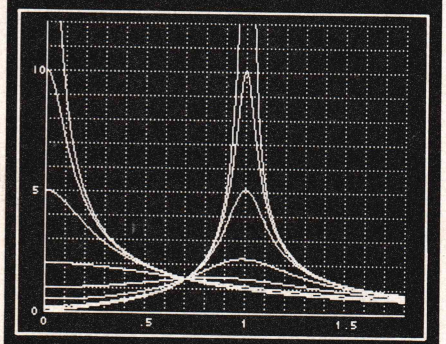
VISA
MASTERCARD

Circle **no. 130** on reader service card.

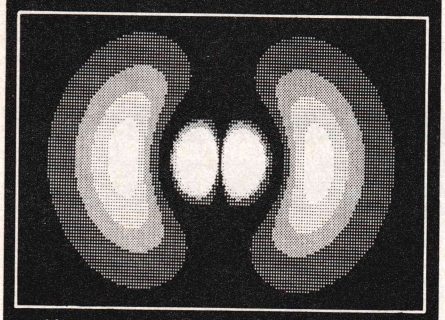
ISYS FORTH

for the Apple®] [

Fixed point speed can rival that of floating point hardware. But the details have been a well kept secret—until now. The following graphs were generated by fixed point examples from the ISYS FORTH manual.



Parallel Resonance with Damping
BASIC 213 sec ISYS FORTH 27 sec



Hydrogen 3p Orbital Cross-section
BASIC 492 sec ISYS FORTH 39 sec

- **Fast** native code compilation. Sieve benchmark: 33 sec
- **Floating Point**—single precision with trancendentals
- **Graphics**—turtle & cartesian with 70-column character set
- **Double Precision** including D*/
- **DOS 3.3 Files** read & written
- **FORTH-83** with standard blocks
- **Full-Screen Editor**
- **Formatter** for word processing
- **Macro Assembler**
- **Price: $99,** no extra charges

ILLYES SYSTEMS
PO Box 2516, Sta A
Champaign, IL 61820

Technical Information:
217/359-6039, mornings

For any Apple] [model, 48K or larger. Apple is a registered trademark of Apple Computer.

Circle **no. 46** on reader service card.

Protocols Listing

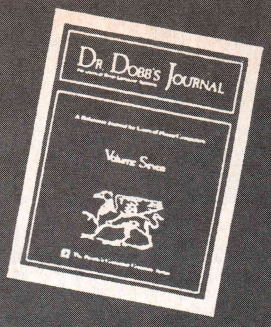
(Listing continued, text begins on page 66)

```
        for( i = 0 ; i ( NAMESIZE ; i++ )/* blank fill name      */
                fcb[ i ] = ' ';
        if( inspec[ 1 ] == ':' )          /* check for drivspec   */
                inptr = 2;                /* point past drivespec */
        else
                inptr = 0;                /* index to start       */
        i = 0;                            /* pointer into fcb     */
        while( TRUE )
                switch( inspec[ inptr++ ] )
                {
                        case '\0':        /* end of input spec    */
                                fcb[ NAMESIZE ] = 0;
                                return( fcb );
                        case '.':         /* extension spec'ed    */
                                i = NAMESIZE - 3; /* extension  */
                                break;
                        default:
                                if( i ( NAMESIZE )
                                        fcb[ i++ ] = toupper(
                                        inspec[ inptr - 1 ] );
                }
}


/*-----------------------------------------------------------------
fillbuf() loads the I/O buffer with a record from the input file.
Returns TRUE if data was available, FALSE if no data left.

References globals: buffer, fd;
Modifies globals: buffer;
-----------------------------------------------------------------*/

int fillbuf()

{
   register int i;                /* scratch counter              */
   int errorchk;                  /* holds EOF in Eco C           */

        for( i = 0 ; i ( RECSIZE ; i++ )
        {
                if( (errorchk = getc( fd)) == EOF )
                        break;
                buffer[ i ] = errorchk;
        }
        if( i == 0 )                          /* no data read     */
                return( FALSE );
        for( ; i ( RECSIZE ; i++ )
                buffer[ i ] = 0;              /* zero fill at EOF */
        return( TRUE );
}


/*-----------------------------------------------------------------
unparse() reassembles a filename from a CP/M fcb into "normal" or
cmpressed form, so that our C functions can deal with them.
Inputs are a pointer to a string to receive the name (name), and
a pointer to a CP/M style fcb entry (buf).
-----------------------------------------------------------------*/

char *unparse( name, buf )
   char *name, *buf;
{
   register int i;                /* 'name' index                 */
   int j;                         /* 'buf' index                  */

        i = 0;                    /* 'driveless' name     */
        for( j = 0 ; j ( NAMESIZE ; j++ )/* transfer chars       */
        {
                if( buf[ j ] != ' ' )   /* (skip spaces)          */
                        name[ i++ ] = buf[ j ] & 0x7F;
                if( j == NAMESIZE-3 )   /* don't forget dot       */
                        name[ i++ ] = '.';
        }
        name[ i ] = '\0';                     /* terminate string */
                                              /* eat terminal dot */
        if( *(index( name, '.' ) + 1) == '\0' )
                *(index( name, '.' )) = '\0';
        return( name );                       /* return pointer   */
}


/*-----------------------------------------------------------------
clrcrc() clears the crc accumulator. Not much to it, actually.
```

(Continued on page 86)

Announcing
BOUND VOLUME 7

Every 1982 Issue Available For Your Personal Reference.



Vol. 7 1982

In 1982 we introduced several significant pieces of software, including the RED text editor and the Runic extensible compiler, and we continued to publish utility programs and useful algorithms. Two new columns, The CP/M Exchange and The 16-Bit Software Toolbox, were launched, and we devoted special issues to FORTH and telecommunications. Resident Intern Dave Cortesi supplied a year of "Clinic" columns while delivering his famous review of JRT Pascal and writing the first serious technical comparison of CP/M-86 and MSDOS. This was also the year we began looking forward to today's generation of microprocessors and operating systems, publishing software for the Motorola 68000 and the Zilog Z8000 as well as Unix code. And in December, we looked beyond, in the provocative essay, "Fifth-generation Computers."

Vol. 1 1976

The material brought together in this volume chronicles the development in 1976 of Tiny BASIC as an alternative to the "finger blistering," front-panel, machine-language programming which was then the only way to do things. This is always pertinent for bit crunching and byte saving, language design theory, home-brew computer construction and the technical history of personal computing.

Topics include: Tiny BASIC, the (very) first word on CP/M, Speech Synthesis, Floating Point Routines, Timer Routines, Building an IMSAI, and more.

Vol. 2 1977

1977 found **DDJ** still on the forefront. These issues offer refinements of Tiny BASIC, plus then state-of-the-art utilities, the advent of PILOT for microcomputers and a great deal of material centering around the Intel 8080, including a complete operating system. Products just becoming available for reviews were the H-8, KIM-1, MITS BASIC, Poly Basic, and NIBL.

Articles are about Lawrence Livermore Lab's BASIC, Alpha-Micro, String Handling, Cyphers, High Speed Interaction, I/O, Tiny Pilot & Turtle Graphics, many utilities, and even more.

Vol. 3 1978

The microcomputer industry entered its adolescence in 1978. This volume brings together the issues which began dealing with the 6502, with mass-market machines and languages to match. The authors began speaking more in terms of technique, rather than of specific implementations; because of this, they were able to continue laying the groundwork industry would follow. These articles relate very closely to what is generally available today.

Languages covered in depth were SAM76, Pilot, Pascal, and Lisp, in addition to RAM Testers, S-100 Bus Standard Proposal, Disassemblers, Editors, and much, much more.

Vol. 4 1979

This volume heralds a wider interest in telecommunications, in algorithms, and in faster, more powerful utilities and languages. Innovation is still present in every page, and more attention is paid to the best ways to use the processors which have proven longevity—primarily the 8080/Z80, 6502, and 6800. The subject matter is invaluable both as a learning tool and as a frequent source of reference.

Main subjects include: Programming Problems/Solutions, Pascal, Information Network Proposal, Floating Point Arithmetic, 8-bit to 16-bit Conversion, Pseudo-random Sequences, and Interfacing a Micro to a Mainframe—more than ever!

Vol. 5 1980

All the ground-breaking issues from 1980 in one volume! Systems software reached a new level with the advent of CP/M, chronicled herein by Gary Kildall and others (**DDJ's** all-CP/M issue sold out within weeks of publication). Software portability became a topic of greater import, and **DDJ** published Ron Cain's immediately famous Small-C compiler—reprinted here in full!

Contents include: The Evolution of CP/M, a CP/M-Flavored C Inerpreter, Ron Cain's C Compiler for the 8080, Further with Tiny BASIC, a Syntax-Oriented Compiler Writing Language, CP/M to UCSD Pascal File Conversion, Run-time Library for the Small-C Compiler and, as always, even more!

Vol. 6 1981

1981 saw our first all-FORTH issue (now sold out), along with continuing coverage of CP/M, small-C, telecommunications, and new languages. Dave Cortesi opened "Dr. Dobb's Clinic" in 1981, beginning one of the magazine's most popular features.

Highlights: information on PCNET, the Conference Tree, and The Electric Phone Book, writing your own compiler, a systems programming language, and Tiny BASIC for the 6809.

YES! ☐ Please send me the following Volumes of **Dr. Dobb's Journal.**
☐ ALL 7 for ONLY $165, a savings of over 15%!

Payment must accompany your order.

Please charge my: ☐ Visa ☐ MasterCard ☐ American Express
I enclose ☐ Check/money order

Card #_____ Expiration Date_____

Signature_____

Name_____ Address_____

City_____ State_____ Zip_____

Mail to: Dr. Dobb's Journal, 2464 Embarcadero Way, Palo Alto, CA 94303
Allow 6–9 weeks for delivery.

Vol. 1_____ x $26.75 = _____
Vol. 2_____ x $27.75 = _____
Vol. 3_____ x $27.75 = _____
Vol. 4_____ x $27.75 = _____
Vol. 5_____ x $27.75 = _____
Vol. 6_____ x $27.75 = _____
Vol. 7_____ x $30.75 = _____
All 7 _____ x $165.00 = _____

Sub-total $_____

Postage & Handling _____
Must be included with order.
Please add $1.25 per book in U.S. ($3.25 each surface mail outside U.S. Foreign Airmail rates available on request.)

TOTAL $_____

LISP
FOR THE IBM PERSONAL COMPUTER.

THE PREMIER LANGUAGE OF ARTIFICIAL INTELLIGENCE FOR YOUR IBM PC.

- **DATA TYPES**
  - Lists and Symbols
  - Unlimited Precision Integers
  - Floating Point Numbers
  - Character Strings
  - Multidimensional Arrays
  - Files
  - Machine Language Code
- **MEMORY MANAGEMENT**
  - Full Memory Space Supported
  - Dynamic Allocation
  - Compacting Garbage Collector
- **FUNCTION TYPES**
  - EXPR/FEXPR/MACRO
  - Machine Language Primitives
  - Over 190 Primitive Functions
- **IO SUPPORT**
  - Multiple Display Windows
  - Cursor Control
  - All Function Keys Supported
  - Read and Splice Macros
  - Disk Files
- **POWERFUL ERROR RECOVERY**
- **8087 SUPPORT**
- **COLOR GRAPHICS**
- **LISP LIBRARY**
  - Structured Programming Macros
  - Editor and Formatter
  - Package Support
  - Debugging Functions
  - .OBJ File Loader
- **RUNS UNDER PC-DOS 1.1 or 2.0**

IQLISP

5¼" Diskette and Manual _____ $175.00
Manual Only _____ $ 30.00

Integral Quality

P.O. Box 31970
Seattle, Washington 98103-0070
(206) 527-2918

Washington State residents add sales tax.
VISA and MASTERCARD accepted.
Shipping included for prepaid orders.

Protocols Listing

(Listing continued, text begins on page 66)

```
References globals:
Modifies globals: crcaccum, checksum;
-----------------------------------------------------------------*/

VOID clrcrc()

{
        crcaccum =
        checksum = 0;
}

/*-----------------------------------------------------------------
updcrc() updates the crc accumulator, if 'crc' is TRUE, else
updates the checksum.
'x' is the byte to be added to CRC or checksum.
CCITT polynomial.

References globals: crc;
Modifies globals: crcaccum, checksum;
-----------------------------------------------------------------*/

VOID updcrc( x )
   char x;
{
   unsigned shifter, i, flag;

        if( crc )
        {
                for( shifter = 0x80 ; shifter ; shifter >>= 1 )
                {
                        flag = (crcaccum & 0x8000);
                        crcaccum <<= 1;
                        crcaccum |= ((shifter & x) ? 1 : 0);
                        if( flag )
                                crcaccum ^= 0x1021;
                }
        }
        else
                checksum += x;
}


/*-----------------------------------------------------------------
abort() scans console for ^X abort code. Exit via error().
System Dependent. If not supportable, convert to null function
or delete and use a preprocessor macro to make it go away.
-----------------------------------------------------------------*/

VOID abort()
{
        if( _bdos( 6, 0xFF ) == 'X'-'@' )
                error( "\rOperator requested abort" );
}


/*-----------------------------------------------------------------
error() does error abort cleanup

References globals: fd;
-----------------------------------------------------------------*/

VOID error( str )
   char *str;
{
        fclose( fd );                   /* dump any open files  */
        tx( CAN );                      /* alert remote         */
        putchar( BELL );                /* alert operator       */
        printf("\n\nError - %s\nPress any key to continue", str );
        getchar();                      /* pause                */
        longjmp( &to_menu, 0 );         /* rejoin menu          */
}


/*-----------------------------------------------------------------
sleep() does a short delay to account for transmission line
latency, etc.
-----------------------------------------------------------------*/

VOID sleep()
{
   register unsigned int i;

        for( i=0 ; i < MAGIC_NUMBER ; i++ )
                ;
}
```

(Continued on page 88)

APC MEGABASIC

'The Mercedes-Benz' of BASICs

8086/88 IBM PC AT MS-DOS
CP/M-86 MP/M-86 TURBODOS

NETWORK COMPATIBLE: PCNET, 3-COM, NOVELL, ETC.

MEGABASIC reduces program development time and memory requirements dramatically, executes up to 6 times faster than MBASIC interpreter, is **highly portable among virtually all microcomputers,** and is supported by outstanding documentation.

BENEFITS:

- Addresses up to 1 Mb programs, data.
- Executes as fast as many compilers.
- Superior debugging facilities.
- BCD arithmetic eliminates rounding errors.
- Simple to use—No complicated field statements.
- Source code protection with "scramble" utility.
- Easy-to-use strings (up to 64K long).

THE COMPLETE PACKAGE:

—Developmental version of MEGABASIC in precisions up to 18 digits.
—Run-time semi-compiler version.
—Compaction utility reduces program size.
—Cross-reference generator lists all variables, arrays, subroutines, functions, etc.
—Function library with fast sorts, yes/no prompt routines, matrix manipulation and other routines ready to plug into your programs.
—Configuration program.
—380-page manual with more than 2,500 index entries.
Complete package: $400, with *30-day money back guarantee.*

AMERICAN PLANNING CORPORATION
Suite 425
4600 Duke Street
Alexandria, VA 22304
1-800-368-2248
(In Virginia, 1-703-751-2574)

Dealer inquiries invited. VISA or MasterCard accepted.

Circle **no. 8** on reader service card.

Fortran Scientific Subroutine Package

Contains Approx. 100 Fortran Subroutines Covering:

1. Matrix Storage and Operations
2. Correlation and Regression
3. Design Analysis
4. Discriminant Analysis
5. Factor Analysis
6. Eigen Analysis
7. Time Series
8. Nonparametric Statistics
9. Distribution Functions
10. Linear Analysis
11. Polynomial Solutions
12. Data Screening

Sources Included

$295.00

FORLIB-PLUS™

Contains three assembly coded LIBRARIES plus support, FORTRAN coded subroutines and DEMO programs.

The three LIBRARIES contain support for GRAPHICS, COMMUNICATION, and FILE HANDLING/DISK SUPPORT. An additional feature within the graphics library is the capability of one fortran program calling another and passing data to it. Within the communication library, there are routines which will permit interrupt driven, buffered data to be received. With this capability, 9600 BAUD communication is possible. The file handling library contains all the required software to be DOS 3.0 PATHNAME compatible.

STRINGS & THINGS™

Support for CHARACTER MANIPULATION (string support), SHELL, BATCH, MUSIC, CMD LINE, and ENVIRON CTRL.

$69.95 each

P.O. Box 2517
Cypress, CA 90630

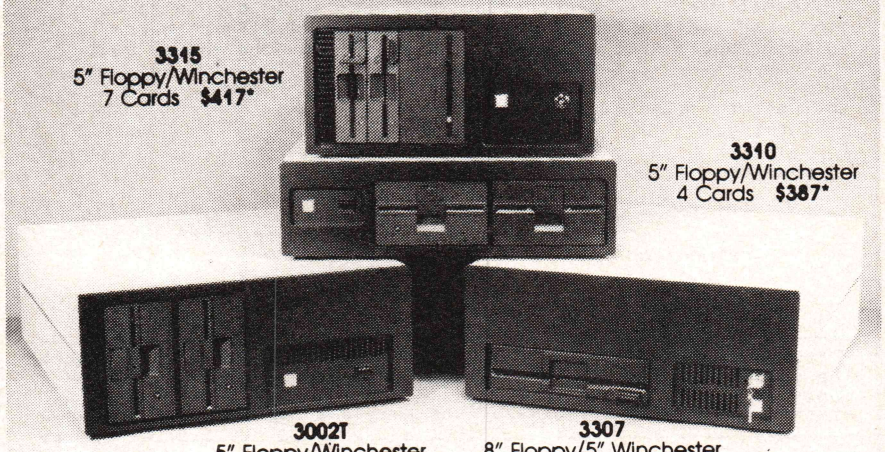
ALPHA COMPUTER SERVICE **(714) 894-6808**

California residents, please add 6% sales tax.

Versions available for IBM Professional Fortran or MICROSOFT 3.2 Fortran

Circle **no. 1** on reader service card.

Classy Chassis



3315
5" Floppy/Winchester
7 Cards **$417***

3310
5" Floppy/Winchester
4 Cards **$387***

3002T
5" Floppy/Winchester
10 Cards **$565***

3307
8" Floppy/5" Winchester
7 Cards **$494***

laser 3000 MAIN/FRAMES & DISC ENCLOSURES FROM $100

LASER 3000 DISC/COVERS (not shown)

3916F
5" Floppy **$100***

3915
2 ea. 5" Winchester **$199***

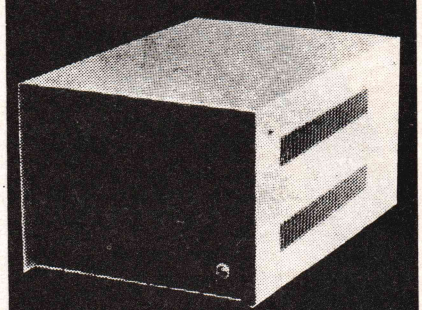
(Disk drives not included)

*1 piece; prices lower in quantity.

INTEGRAND RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291 209/651-1203

AMPRO "Little Board" MAIN/FRAMES

6 Models from **$125***



$150 (1 piece*)
MODEL 2800
Includes power supply & fan
(Disk Drives and Little Board not included)

AMPRO & Little Board are ™ AMPRO computers

INTEGRAND RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291
209/651-1203

Circle **no. 15** on reader service card.

NOW C HERE!
CROSS SOFTWARE for the NS32000

Also Available for IBM PC

INCLUDES:

★ Cross Assembler ★
★ Cross Linker ★
★ Debugger ★
★ N.S. ISE Support ★
★ Librarian ★
★ Pascal Cross Compiler ★
★ C Cross Compiler ★

U.S. prices start at $500

SOLUTIONWARE
1283 Mt. View-Alviso Rd.
Suite B
Sunnyvale, Calif. 94089
408/745-7818 • TLX 4994264

Circle **no. 118** on reader service card.

FORTRAN PROGRAMMERS

Discover why you should be using
F77L
the complete implementation of the ANSI FORTRAN 77 Standard for the IBM PC and compatibles.
If you are serious about your FORTRAN programming, you should be using F77L.
$477

Lahey Computer Systems, Inc.
31244 Palos Verdes Drive West, Suite 243
Rancho Palos Verdes, California 90274
(213) 541-1200
Serving the FORTRAN community since 1969

Circle **no. 104** on reader service card.

YOU CAN C

VIEW v3
The ultimate CP/M disk utility

- **Edits** any disk sector
- **Displays** allocation bit map
- **Rebuilds** files automatically from selected blocks

If you ever had to recover a crashed disk, you need **VIEW**. Use **VIEW** to patch damaged sectors, un-erase files, examine other disk formats, more.

COMPLETE C SOURCE CODE $59

Other C Products
EZZAP, ICX, MEDIT, more.
Send for a catalog today.

CP/M®, Digital Research
ISIS-II ®, Intel Corp.

Western Wares 303-327-4898
Box C • Norwood, CO 81423

Circle **no. 59** on reader service card.

Protocols Listing

(Listing continued, text begins on page 66)

```
/*-------------------------------------------------------------
wait() waits for a character - timeout built in. Timeout
condition causes error return to menu.
Input parameter 'time' controls duration of wait.
-------------------------------------------------------------*/

int wait( time )
   int time;
{
   register unsigned int i;              /* loop timer          */
   int j;                                /* timeout count       */

        j =                              /* timeout count       */
        i = 0;                           /* inner timeout count */
        while( !rxstat() )
                if( i++ ) MAGIC_NUMBER * 3 )/* about 1.5 secs   */
                        if( !(time--) ) /* check retry count   */
                                error( "Receiver timed out" );
                        else             /* more tries available */
                        {
                                abort(); /* scan for ^X         */
                                if( !j ) /* \n first timeout    */
                                        printf( "\n" );
                                printf( "\rtimeout %d   ", ++j );
                                i = 0;
                        }
        return( rx() );                  /* send back char      */
}

/*-------------------------------------------------------------
swait() does a short wait for a char from rx, returns ERROR on
a timeout, or char received.
-------------------------------------------------------------*/

int swait()
{
   register unsigned int i;              /* loop timer          */

        i = 0;
        while( i++ < MAGIC_NUMBER )      /* loop timing test    */
                if( rxstat() )           /* anything yet?       */
                        return( rx() ); /* yes, return char    */
        return( ERROR );                 /* timeout exit        */
}


/*-------------------------------------------------------------
waitcan() waits for a character - If recv's CAN aborts to main
menu. Uses wait() for timing loop, so timeout also returns
to main menu. Input argument 'time' controls duration of wait.
-------------------------------------------------------------*/

int waitcan( time )
   int time;
{
   char ch;

        if( (ch = wait( time )) == CAN )
                error( "Received cancellation request" );
        return( ch );
}


/*-------------------------------------------------------------
=============================================================
The following functions are machine dependant and should be
modified to run on your machine.
=============================================================
-------------------------------------------------------------*/


/*-------------------------------------------------------------
init() initializes globals. In addition, UART initialization
goes here as well, if required. No input parameters. May have
some machine dependant stuff here.

References globals:
Modifies globals: crc, batch, buffer, line;
-------------------------------------------------------------*/

VOID init()
{
        buffer = alloc( RECSIZE );       /* transfer buffer     */
```

```
        line = alloc( 81 );             /* scatch input line    */
        crc =                           /* CRC mode             */
        batch = TRUE;                   /* not really required  */
}


/*-----------------------------------------------------------------
rx() receives a char from uart. No timeout checking is performed.
Also, no status check is done. Bad news if you call this without
checking status first... Done that way to speed up things.

                        ZORBA Version
-----------------------------------------------------------------*/

int rx()
{
        return( inp( RXDATA ) );
}


/*-----------------------------------------------------------------
rxstat() returns TRUE if char waiting for input.

              ZORBA Version (interrupt fifo)
-----------------------------------------------------------------*/

int rxstat()
{
        return( inp( UARTCMD ) & 2 );
}


/*-----------------------------------------------------------------
tx( ch ) transmits the char 'ch', and calls the CRC/checksum
update. A timeout is built in for transmission faults.

                        ZORBA Version
-----------------------------------------------------------------*/

VOID tx( ch )
   char ch;
{

   register unsigned int i;             /* timing loop counter  */

        i = 0;
        while( !txstat() )              /* ready?               */
                if( i++ > MAGIC_NUMBER )/* loop about 3 secs    */
                        error( "UART not ready??" );
        outp( TXDATA, ch );             /* to UART              */
        updcrc( ch );                   /* stir CRC a little    */
}


/*-----------------------------------------------------------------

txstat() returns TRUE if tx register is empty

                        ZORBA Version
-----------------------------------------------------------------*/

int txstat()
{
        return( inp( UARTCMD ) & 4 );
}


/*-----------------------------------------------------------------
quit() does exit processing.
-----------------------------------------------------------------*/

VOID quit()
{
        exit( 0 );
}
```

End Listing

Demo Disk Call (800) 327-5895

VEDIT PLUS

The First Multiple File Programmable Editor

VEDIT has been acclaimed for the last five years as the Industry Standard in text editing.

InfoWorld
Software Report Card

VEDIT 1.36

	Poor	Fair	Good	Excellent
Performance	☐	☐	☐	☑
Documentation	☐	☐	☐	☑
Ease of Use	☐	☐	☑	☐
Error Handling	☐	☐	☐	☑

Now there's VEDIT PLUS.

VEDIT PLUS is the ideal tool for program editing and technical writing. It gives you every editing function you expect, plus it can:

- Edit multiple files of any size
- Compare files
- Be fully customized
- Perform arithmetic computations
- Be expanded with print formatting & spell checking-correcting

The power of VEDIT PLUS lets you increase your productivity with:

- Numeric, relational, and logical functions
- If-Then-Else decision making
- Easy file handling
- The ability to create prompts for user input and menus
- Special programming features

Expect a lot from VEDIT PLUS. It's small (23K), fast and it's from CompuView - nationally recognized for user support.

VEDIT PLUS $225
VEDIT $150

CompuView
PRODUCTS, INC.

1955 Pauline, Ann Arbor, MI 48103
(313) 996-1299 - (800) 327-5895

Circle **no. 9** on reader service card.

Slash Programming Time in Half!

With FirsTime™

- Fast program entry through single keystroke statement generators.
- Fast editing through syntax oriented cursor movements.
- Dramatically reduced debugging time through immediate syntax checking.
- Fast development through unique programmer oriented features.
- Automatic program formatter.

FirsTime is a true syntax directed editor.

FirsTime ensures the integrity of your programs by performing all editing tasks like moves, inserts and deletes along the syntactic elements of a program. For example, when you move an IF statement, FirsTime will move the corresponding THEN and ELSE clauses with it.

Even FirsTime's cursor movements are by syntax elements instead of characters. The cursor automatically skips over blank spaces and required keywords and goes directly to the next editable position.

FirsTime is a Syntax Checker

FirsTime checks the syntax of your program statements, and also:

- Semantics like undefined variables and mismatched statement types.
- The contents of include files and macro expansions.
- Statements for errors as they are entered and warns you immediately.

FirsTime is a Program Formatter

FirsTime automatically indents statements as they are entered, saving you from having to track indentation levels and count spaces.

FirsTime has Unique Features

No other editor offer these features:

The *Zoom command* gives you a top down view of your program logic.

The *View command* displays the contents of include files and macro expansions. This is valuable to sophisticated programmers writing complex code or to those updating unfamiliar programs.

FirsTime's *Transform command* lets you change a statement to another similar one with just two keystrokes. For example, you can instantly transform a FOR statement into a WHILE statement.

The *Move at Same Level command* moves the cursor up or down to the next statement at the same indentation level. This is very useful. For example, you can use it to locate the ELSE clause that corresponds to a given THEN clause or to traverse a program one procedure at a time.

FirsTime is Unparalleled

FirsTime is the most advanced syntax directed editor available. It makes programming faster, easier and more fun.

FirsTime for Turbo Pascal $74.95

FirsTime for MicroSoft & IBM Pascal $245

FirsTime for C $295

TO ORDER CALL (201) 741-8188

or write:

Spruce Technology Corporation

189 E. Bergen Place
Red Bank, NJ 07701

In Germany, Austria and Switzerland contact:

Markt & Technik Software Verlag
Munchen, W. Germany
(089) 4613-0

FirsTime is a trademark of Spruce Technology Corporation • MS-DOS is a trademark of Microsoft Corporation
IBM is a trademark of International Business Machines, Inc. • Turbo Pascal is a trademark of Borland International

DSD 80

FULL SCREEN SYMBOLIC DEBUGGER

"THE SINGLE BEST DEBUGGER FOR CP/M-80. A TRULY AMAZING PRODUCT."

LEOR ZOLMAN
AUTHOR OF BDS C

Complete upward compatibility with DDT
Simultaneous instruction, register, stack & memory displays
Software In-Circuit-Emulator provides write protected memory, execute only code and stack protection.
Full Z80 support with Intel or Zilog Mnemonics
Thirty day money back guarantee
On-line help & 50 page user manual

NOW ONLY $125.

SOFTADVANCES

P.O. BOX 49473 AUSTIN, TEXAS 78765 (512) 478-4763

MasterCard VISA

Circle **no. 63** on reader service card.

You read Dr. Dobb's Journal And You Don't Subscribe?!

Save over $23.00 off newstand prices for 2 yrs.
Save over $10.00 for 1 yr.

Can you afford to miss an issue with information vital to your interests? As a subscriber you can look forward to articles on Small-C, FORTH, CP/M, S-100, Compiler optimization, Concurrent Programming and more, delivered right to your door. And you'll never miss the issue that covers your project.

Yes! Sign me up for
___ 2 yrs. $47 ___ 1 yr. $25

___ I enclose a check/money order
___ Charge my Visa, MasterCard, American Express
___ Please bill me later

Name _____
Address _____
_____
_____ Zip _____
Credit Card _____ Exp. date _____
Account No. _____
Signature _____

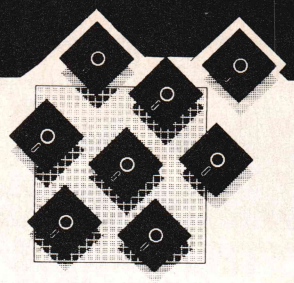
Mail To: Dr. Dobb's Journal, PO Box 27809, San Diego, CA 92128 3041

Is $19.97 too much to spend to avoid a million dollar mistake?

Business Software magazine is a valuable resource that helps you squeeze every penny's worth of potential out of your computer system. It is written for the business person and decision maker who wants practical, "hands-on" software solutions to real problems, written in a straightforward, jargon-free language. Every month, **Business Software** brings you: case studies of business computer applications; company profiles, including who is using what software and why; software tutorials that explain what the manual left out; information on the latest products, as well as the longtime favorites, all from a business perspective.

To subscribe, please fill out this coupon and return it to:

Business Software,
P.O. Box 27975,
San Diego, CA 92128
Name _____
Address _____
City _____
State _____ Zip _____
Bill my ☐ VISA ☐ M/C
☐ Bill me later ☐ $19.97
Card No. _____
Expiration date _____
Signature _____
Please allow 6–9 weeks for address change to take effect 3033



Envoy, Version 1.1

Company: Artisoft Inc., 2450 E. Speedway, Suite 4, Tucson, AZ 85719 (602) 327-4305
Computer: IBM PC family, 16-bit compatibles, 8-bit CP/M
Price: $49.95
Circle Reader Service No. 144
Reviewed by David W. Carroll

The trend in microcomputer software is toward high-volume, high-quality, low-cost software (under $100). Borland International pioneered this "volks-software" concept with Turbo Pascal and Sidekick, and other companies are now following this path.

Commercial telecommunications software typically costs $200 and more, but Artisoft of Tucson, is now offering its Envoy communications package with versions that run on almost any microcomputer system for only $49.95.

"With over a dozen major commercial programs available, who needs another?" you might well ask. First, the price of Envoy represents a breakthrough. Second, the program is small, fast, and easy to use. Third, Envoy offers some unique features. Finally, compatible versions are available for almost any type of microcomputer and operating system.

Operation

Envoy is a menu-driven communications program. This means that it has two modes: command and terminal. In command mode, the current menu is shown on the screen. In terminal mode, you are on line with your serial port or modem, and Envoy is transparent to all keyboard input except for your terminal mode "exit" character, usually escape <Esc>. Envoy requires you to set your terminal mode exit character when you start each session. The program's main menu appears below:

```
*** Main Menu ***

<T>   — Smart Terminal
<E>   — ENVOY file transfers
<X>   — XMODEM file transfers
<P>   — Port Parameters
<U>   — Utilities
<Ret> — Previous menu
<Esc> — Exits ENVOY

Enter your selection –>
```

The program offers dumb terminal operation, ASCII file send and buffer capture, and protocol file transfers. Envoy supports both the Ward Christensen XMODEM protocol and the little known ANSI standard X3.28 file transfer protocol.

The ANSI protocol is designed so that either computer in a data link can act as master while the other is a slave. This allows multiple file transfers and remote system utility operation (you can use the remote system's utility menu). The transfer protocol is faster than XMODEM and provides greater error checking.

This menu appears when you select Smart Terminal:

```
*** Smart Terminal ***

Capture Buffer is Off

<T>   — Terminal Mode
<S>   — Send Text File
<M>   — Modem Functions
<O>   — Terminal Mode Options
<C>   — Capture Buffer On
<B>   — Buffer Utilization
<W>   — Write Buffer to Disk
<D>   — Display Buffer
<U>   — Utilities
<Ret> — Previous menu
<Esc> — Exits ENVOY

Enter your selection –> B

Capture Buffer 57362 bytes
Remaining Space 57362 bytes

[Enter any key to continue]
```

Envoy uses an autodial menu and even supports simple "script" files for autologon. It will support almost any modem because of its free-form modem command/phone number directory. The directory consists of a file with entries created using a word processor or editor (like WordStar or EDLIN). A few typical directory entries for a Hayes Smartmodem follow:

```
Redial [A/|]
Disconnect [_+++_ATH0]
Information [ATDT1-555-1212]
```

Here is a sample directory listing:

```
*** Modem Functions ***

<A>   — Redial
<B>   — Disconnect
<C>   — Information
<D>   — ARTISOFT, Inc.
<E>   — CompuServe
<Ret> — Previous menu
<Esc> — Exits ENVOY

Enter your selection –>
```

Envoy scripts can include keyboard control characters, delays, and character-matching strings. The special characters ^, _, and | provide these features when inserted in script entries. Question marks indicate strings to match.

You can set the terminal mode

Copyright by David W. Carroll

communications parameters, which include stripping line feeds, local echo mode, and delay of up to 9 sec between each line sent for ASCII text file transfers:

*** Terminal Mode Options ***

<L> — Send LFs On
<D> — Delay Lines (0)
<E> — Local Echo On
<X> — Change Exit Key
<Ret> — Previous menu
<Esc> — Exits ENVOY

Enter your selection –>

You may also set port parameters (speed, parity, etc.) from within Envoy.

Full operating system utilities are available from within Envoy, and PC/MSDOS users have full access to DOS 2.0 subdirectory commands and paths. The program even supports remote system operation (unattended). This menu is for the IBM PC version of Envoy PC:

*** Utilities ***

<D> — Directory
<L> — Log to another disk
<E> — Erase files
<T> — Type files
<C> — Copy a file
<R> — Rename a file
<P> — Print files
<N> — Change directory path
<S> — Create a directory path
<X> — Remove a directory path
<Ret> — Previous menu
<Esc> — Exits ENVOY

Enter your selection –>

Envoy is written in assembly code; that means it is fast and compact. The program takes a total of 7–9K disk space, depending on the version. That is small enough to put on every floppy—not much more space than the old standby MODEM7 program requires—and Envoy offers many more features.

Evaluation

Envoy PC operates reliably in all modes. My only complaints were regarding the effort required to make new directory entries. I found it clumsy to exit the program, start up a word processor, and enter numbers in the directory file. The lack of a real-time directory entry mode, similar to what Crosstalk and Smartcom II offer, was irritating. I also didn't like to reenter the required modem commands in each directory entry, though this is a minor point.

Overall, the Envoy communications programs are well designed, easy to use, and versatile. For a reasonable price, they provide virtually every communications feature you could require.

DDJ

DEZIGN

THE STRUCTURED PROGRAMMING TOOL FOR MODERN TIMES

- Design your programs right on the screen, using modern techniques based on the popular Jackson Structured Programming method (JSP)!
- **DEZIGN** is more than just another flowcharting tool. It is an integrated tool for designing and documenting programs and for generating ADA, C, PASCAL, and PL/I source code, as well as dBASE II and dBASE III command files.
- **DEZIGN** enables you to create Data and Program Structure Diagrams using the Sequence, Selection (IF-THEN-ELSE), and Iteration (DO WHILE) constructs; assign detailed statements to the diagrams; and synthesize source code from the control logic represented on the diagrams and the detailed statements assigned to them.
- **DEZIGN** lists for $200. It runs on the IBM PC, XT, or AT and requires 128K RAM, one disk drive, and an 80-column color or monochrome display.
  - DEZIGN-PC runs under DOS 2.0, 2.1, and 3.0.
  - DEZIGN-86 runs under CP/M-86 1.1.
- Want to learn more? Please contact us concerning pricing and availability of JSP reference texts and seminars.

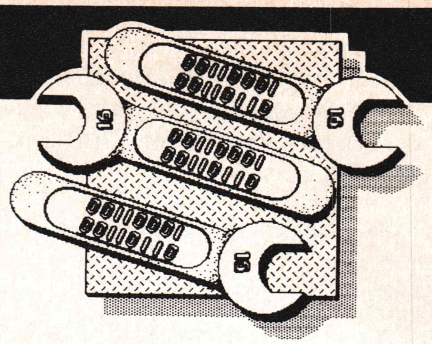
ZEDUCOMP • P.O. BOX 68 • STIRLING, NJ 07980

(201) 755-2262

dBASE II and dBASE III are trademarks of Ashton-Tate, Inc.

Circle **no. 126** on reader service card.

16-BIT SOFTWARE TOOLBOX



by Ray Duncan

IBM PC WordStar and the HP LaserJet

We at Laboratory Microsystems recently purchased an HP LaserJet printer to produce our various software manuals. After buying the LaserJet, we were dismayed to find that it is essentially impossible to configure IBM PC WordStar to properly use the printer's boldface, italics, and various fonts. WordStar allows the user to patch in only four-character escape sequences, while nearly every important LaserJet control sequence is five bytes or longer. When we contacted the HP dealer about this problem, his response was, "Well, you'll just have to switch to Spellbinder; it supports all the LaserJet capabilities."

Because we have around 10 Mb of documentation files that we would have to convert if we switched word processors, this option didn't sound too attractive. My first attempt at getting around the problem was to write a character filter that read a WordStar document file, translated the various control codes (such as ^B) into the proper LaserJet escape sequences, and sent the result to the printer. This worked acceptably as far as print quality, but it required an exit from WordStar whenever you wanted to print something, which made WordStar's capability of background printing while editing useless.

It began to look as if we would have to face the facts and do some heavy-duty disassembly and patching of WordStar. On a long shot, I dropped a note to MicroPro Technical Support, asking them if they knew of anyone who had already worked out the necessary patches. Much to my amazement, I got an immediate phone call from a nice lady at MicroPro who said that a LaserJet version of WordStar was already available to licensed users for $25.00. I duly sent off my check and in a few days received a diskette containing a new WS.OVL file and a batch file that performs some patches to the WS.COM file.

After following the installation steps in the accompanying documentation, I ended up with a WordStar that supports italics, boldface, double-strike, underline, over-strike, micro-justification, superscripting, subscripting, variable character pitch, variable line height, and multiple copies on the HP LaserJet. It's a little slow (only 2–3 pages/min throughput) and has a few bugs (a ^Y at the start of a line makes it go crazy), but it works pretty well. You can get this enhancement disk by sending a polite letter, including your WordStar version 3.3 serial number and $25.00, to LaserJet Technical Support, MicroPro International Corp., 33 San Pablo Ave., San Rafael, CA 94903.

Microsoft 8086 MacroAssembler 2.0

Many moons after Microsoft announced version 2.0 of the 8086 MacroAssembler, it has appeared in the IBM Product Centers in the last few weeks without fanfare. Along with the new, higher version number comes a new, higher price ($175.00). Silly me. I thought that because I had bought version 1 with its zillions of bugs I might be entitled to a reduced cost update, but the Product Center salesman nearly killed himself laughing when I brought *that* up. Finally, gritting my teeth and hoping that the new version would be a little more robust than the old, I hauled out my well-worn MasterCard and took home a copy.

If you choose to go the same route, what will you get for your money? Well, the first thing you get is a MacroAssembler that has an even more severe case of bloat than the first version (the EXE file is 76,544 bytes compared to 67,584 bytes for version 1). The 8087 mnemonics are now fully supported, along with most of the additional 80286 mnemonics that would be used in that CPU's real mode. The performance of the MacroAssembler has improved considerably: a 106K source file (approximately 3200 lines), which required 13 min 57 sec to assemble with version 1, is assembled in 4 min 13 sec with version 2.

Version 2 provides a new version of the Linker (which has also suffered more bloat to the tune of about 6K), a documented Library manager, and a new program called SALUT (Structured Assembly Language Utility), which converts an assembly language source file containing high-level control structures into a source file that can be fed to the assembler. Although IBM pushes SALUT pretty hard in the MacroAssembler manual, I see no evidence in the BIOS listings or distributed PCDOS assembler source files (such as VDISK.ASM) that IBM programmers ever use it. Perhaps IBM is adopting some of Apple's policies for "the rest of us."

The manual for the MacroAssembler has been drastically overhauled. The previous "encyclopedia" format, which devoted a page to each of the assembler directives and 8086 mnemonics with examples, has been abandoned. The new manual is heavily oriented to *operation* of the assembler and associated utilities (everything is called "sessions" now) and provides almost no programming guidance whatsoever. A copy of Rector and Alexy's *The 8086 Book* or the Intel *iAPX 86 User's Manual* will be

indispensable to people who start out with the new MacroAssembler and no previous experience.

Some of the known bugs in version 1 appear to have been fixed; these include the operand order in the SHL and SHR operators, the XOR that used to work like OR, the packed decimal data generated by the DT pseudo-op, the failure of DUP to work properly inside the invocation of a STRUC, and the lack of error messages and generation of "funny" opcodes for some nonexistent instructions (such as CMP ES,0). Error detection (and the quality of the error messages) seems generally improved.

Some bugs, however, remain. I've provided a few examples in Listing Two (page 104). A major problem seems to be inconsistent handling of signed and unsigned values by the logical operators. This is "explained" to some extent by a comment in the manual on page 2-28: "When relational operators are used, results are produced by comparing 17-bit signed numbers, where the 17th bit can contain only a sign bit." This accounts for the behavior of the LT and GT operators, but if it were strictly true, the expression "–1 EQ 0FFFFH" would return a false flag (which it doesn't, as you can see by the listing). Anyway, it's a mystery to me why Microsoft thought it should support within expressions a 17-bit data type that has no physical equivalent in the 8086 processor family. Wouldn't it make more sense to have the assembler evaluate expressions to the same results that you would get with in-line machine code?

The TRIM Filter

In the last few months, we have printed three new "filters" for MSDOS: CLEAN, TEE, and TK. In this month's column, we print another filter, named TRIM, contributed by A. K. Head of Melbourne, Australia (Listing One, page 96). I have taken the liberty of retyping and reformatting the program from his original listing, and I have added some error messages; hence, credit for the concept and implementation should go to Mr. Head, while blame for any bugs or other deficiencies specially introduced by publication in *DDJ* is mine.

In spite of its brevity, TRIM is a fairly sophisticated filter that can perform one of several operations:

- Exclude all the characters *outside* a given range of columns from each line of a file
- Exclude all the characters *inside* a given range of columns from each line of a file
- Delete trailing blanks from each line of a file
- Delete blank lines from a file

Documentation for the use of TRIM appears in the beginning of the source listing itself. You can use TRIM in combination with SORT and various other commands by means of the Unix-like pipes of MSDOS 2.x to perform complex operations. For example, the following command line will isolate the filename and extension from each line of a directory listing, sort the list, and leave the result

Advanced Screen Management made easy

Now a professional software tool from Creative Solutions.

WINDOWS FOR C™

More than a window display system, WINDOWS FOR C is a video tool kit for all screen management tasks.

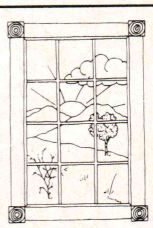
- **Pop-up menus and help files**
- **Auto memory management**
- **Keyboard interpreter**
- **Word wrap**
- **Auto scroll**
- **Highlighting**
- **Color control**
- **Overlay and restore**
- **Plus a library of over 50 building block subroutines**

Designed for enhanced portability. Easy to learn, easy to use.

Once you've tried WINDOWS FOR C, you'll wonder how you ever managed without it.

Full support for IBM PC/XT/AT and compatibles, plus interfaces for non-IBM computers; Lattice C, CI-C86, Mark Wm. C, Aztec C, Microsoft C, DeSmet C (PC/MSDOS),

NEW Ver. 3.1
Enhanced portability.
Topview compatible.
WINDOWS FOR C $195
(specify compiler & version)
Demo disk and manual $ 30
(applies toward purchase)
Full source available.
No royalties.



Creative Solutions
21 Elm Ave., Box T4,
Richford, VT 05476
802-848-7738
Master Card & Visa Accepted
Shipping $2.50
VT residents add 4% tax.

Circle **no. 27** on reader service card.

in a file named SORTED.DIR:

```
A>DIR¦TRIM 1,12¦SORT
    >SORTED.DIR
```

Incidentally, you can download the source for all the programs printed in this column, as well as many other useful MSDOS utilities, from the Laboratory Microsystems Bulletin Board System at (213) 306-3530 (300 or 1200 baud). This BBS is up between 6 PM and 9 AM, Pacific Time, on weekdays and for 24 hours on weekends and holidays. **DDJ**

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 194**.

16-Bit Toolbox (Text begins on page 94)

Listing One

```
        name    trim
        page    55,132
        title   'TRIM - excerpt lines of a file'
;
; TRIM --- excerpts selected columns from each line
;          of a file and writes them to the selected
;          output device or file.
;
; A "filter" for MS-DOS or PC-DOS version 2 or higher,
; after the fashion of Unix.   Reads from the standard input
; (redirectable) and writes to the standard output (redirectable).
; Error messages are directed to the standard error device.
;
; TRIM can be (and usually would be) used in a pipe, e.g.
;
;    | TRIM 7,45 |
;
; transmits only the characters in columns 7 to 45 (inclusive)
; of each line.  A minus sign reverses the action, e.g.
;
;    | TRIM -7,45 |
;
; transmits all characters except those in columns 7 to 45.
;
; Special actions:
;
;    | TRIM 0  |        deletes trailing spaces from lines, and
;    | TRIM -0 |        also discards empty lines
;
; By A. K. Head, 6 Duffryn Place, Melbourne, Australia 3142
; reformatted and error handling added by Ray Duncan
;

command equ     80h             ; buffer for command tail
fcb1    equ     5ch             ; default file control block #1
fcb2    equ     6ch             ; default file control block #2

buflen  equ     16384           ; buffer length, alter to taste

cr      equ     0dh             ; ASCII carriage return
lf      equ     0ah             ; ASCII line feed
ff      equ     0ch             ; ASCII form feed
eof     equ     01ah            ; End-of-file marker
tab     equ     09h             ; ASCII tab code
blank   equ     20h             ; ASCII blank

                                ; DOS 2.x pre-defined handles
stdin   equ     0000            ; standard input file
stdout  equ     0001            ; standard output file
stderr  equ     0002            ; standard error file
stdaux  equ     0003            ; standard auxilliary file
stdprn  equ     0004            ; standard printer file

cseg    segment para public 'CODE'
```

(Continued on page 98)

SMALL C FOR IBM-PC

Small-C Compiler Version 2.1 for PC-DOS/MS-DOS
Source Code included for Compiler & Library
New 8086 optimizations
Rich I/O & Standard Library

$40

CBUG SOURCE LEVEL DEBUGGER FOR SMALL C

Break, Trace, and Change variables all on the source level
Source code included

$40

Datalight
11557 8th Ave. N.E.
Seattle, Washington 98125
(206) 367-1803

ASM or MASM is required with compiler.
Include disk size (160k/320k), and DOS version with order.
VISA & MasterCard accepted. Include card no. & expiration date.
Washington state residents include 7.9% sales tax.
IBM-PC & PC-DOS are trademarks of International Business Machines
MS-DOS is a trademark of Microsoft Corporation.

Circle **no. 6** on reader service card.

$5.00 C Compiler

Due to popular demand, **Dr. Dobb's Journal** has reprinted its most-asked-for C compiler articles by Ron Cain and J. E. Hendrix, each for only $5.00.

Ron Cain's C compiler from sold-out 1980 issues #45 and #48 includes "A Small C Compiler for the 8080s" and "Run-time Library for the Small C Compiler."

The J. E. Hendrix reprint includes part two of "Small-C Compiler v.2" from sold out issue #75 and completes the first part of the compiler article from issue #74 which is included in Dr. Dobb's Bound Volume 7.

To Order: Enclose $5.00 for each copy with this coupon and send to: Dr. Dobb's Journal, 2464 Embarcadero Way, Palo Alto, CA 94303

Please send _____ copy(ies) of the Ron Cain Reprint, and
_____ copy(ies) of the J. E. Hendrix reprint to:

name

address

city state zip

ALL REPRINT ORDERS MUST BE PREPAID. 104

Please allow 6–9 weeks for delivery.

6 TIMES FASTER!

SuperFast Software Development Tools

INCREASE YOUR PROGRAMMING EFFICIENCY
with high-performance software development products from SLR Systems.
No other tools approach the speed or flexibility of the SLR Systems line.

"Z80ASM is an extraordinary product...",
Robert Blum, Sept. 84 DDJ

"...in two words, I'd say speed & flexibility",
Edward Joyce, Nov. 84 Microcomputing

ASSEMBLERS

- RMAC/M80 macros
- Nested INCLUDES & conditionals
- 16 char. labels on externals
- Built in cross-reference
- Optional case significance
- Phase/dephase
- Math on external words and bytes
- Define symbols from console
- Generate COM, HEX, SLR-REL, or Micro-soft-REL files
- Time & Date in listing
- Over 30 configure options

Z80ASM -full Zilog Z80 ..... $125
NEW! Z80ASM+ -all tables virtual $195
NEW! SLRMAC -full Intel 8080, with Z80.LIB extensions internal $125
NEW! SLRMAC+ -all tables virtual $195

Z80 CPU, CP/M compatible, 32K TPA required.

"Z80ASM...a breath of fresh air...",
Computer Language, Feb. 85

MasterCard VISA C.O.D., Check or Money Order Accepted

LINKERS

- Links SLR & M80 format files
- Output HEX or COM file
- Three separate address spaces
- Load map and SID/ZSID .SYM file

SLRNK+ includes:

- All tables overflow to disk
- HEX files do not fill unused space
- Intermodule cross-reference
- EIGHT separate address spaces
- Works with FORTRAN & BASIC
- Generate PRL & SPR files
- Supports manual overlays
- Full 64K output

SLRNK -fastest memory based $125
NEW! SLRNK+ -full featured virtual $195
Combo Paks available from $199. - $299.

For additional information contact SLR Systems
1-800-833-3061, in PA (412) 282-0864
1622 N. Main St., Butler, PA 16001 • Telex 559215

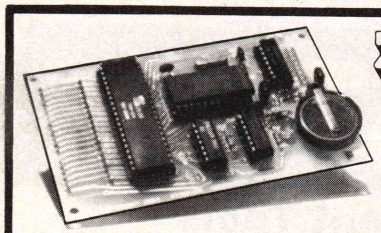
SLR Systems

Circle **no. 78** on reader service card.

Listing One

```
        assume  cs:cseg,ds:cseg

        org     100H                ; start .COM at 100H

start:  jmp     near ptr trim

param1  dw      0                   ; command parameter #1
param2  dw      0                   ; command parameter #2
sign    dw      0                   ; nonzero if "-" in command
count   dw      0                   ; column count, current line
topin   dw      0                   ; chars in input buffer - 1
char    db      0                   ; current character

trim    proc    far

        xor     si,si               ; initialize buffer pointers
        xor     di,di
        mov     bx,fcb1+1           ; addr of parsed parameter 1
        call    getprm              ; convert it
        cmp     ax,0
        je      trunc               ; zero parameter, go truncate
                                    ; trailing blanks etc.
        mov     param1,ax           ; save first parameter
        mov     bx,fcb2+1           ; addr of parsed parameter 2
        cmp     byte ptr [bx],blank ; is it present at all?
        jne     trim0               ; yes, proceed
        jmp     err3                ; no, exit
```

(Continued on page 100)



Ztime-I

CALENDAR/CLOCK

$69 KIT

NOW WITH FILE DATE STAMPING!

- Works with any Z-80 based computer.
- Currently being used in Ampro, Kaypro 2, 4 & 10, Morrow, Northstar, Osborne, Xerox, Zorba and many other computers.
- Piggybacks in Z80 socket.
- Uses National MM58167 clock chip, as featured in May '82 Byte.
- Battery backup keeps time with CPU power off!
- Optional software is available for file date stamping, screen time displays, etc.
- Specify computer type when ordering.
- Packages available:

Fully assembled and tested	$99.
Complete kit	$69.
Bare board and software	$29.
UPS ground shipping	$ 3.

MASTERCARD, VISA, PERSONAL CHECKS, MONEY ORDERS & C.O.D.'S ACCEPTED.

N.Y. STATE RESIDENTS ADD 8% SALES TAX

KCT KENMORE COMPUTER TECHNOLOGIES

P.O. Box 635, Kenmore, New York 14217 (716) 877-0617

Circle no. 30 on reader service card.

C Programmers:
File System Utility Libraries

Source Code Included, No Royalties, Powerful & Portable.

BTree Library **$75.00**

- High speed **random** and **sequential** access.
- **Multiple keys** per data file.
- Up to **16 million** records per file.
- Full documentation and example programs included.

ISAM Driver **$40.00**

- Works with the BTree Library.
- Greatly speeds application development.
- Combines ease of use of **database manager** with flexibility of **programming language.**
- Supports **multi key files** and **dynamic index definition.**
- Very **easy to use; fully documented; example programs included.**

Both products

Are written entirely in **K&R C.**
Come with complete **source code.**
Are **free of any royalty charges.**

+ $3.00 Shipping & Handling Charge.

For more information call:

softfocus

1277 Pallatine Drive
Oakville, Ontario, Canada
L6H 1Z1
(416) 844-2610

Credit cards accepted.

Dealer inquiries invited.

Circle no. 89 on reader service card.

DDJ Classifieds

Software

PCBTAM
Communications Access Method

Allows an IBM PC or compatible to perform BISYNC communication. PCBTAM is a general purpose interrupt driven access method usable by any Microsoft language. Requires IBM BSCA card and PC-DOS.
- Source or object license
- Asynchronous version available (PCATAM)
- Modifiable for other USARTS.

For details write:

SYMBIOTIC
Symbiotic Protocol Converters, Inc.
1011 Clifton Avenue, Clifton, New Jersey 07013
(201) 777-8454

DriveLiner

Drive Alignment Test Program for CP/M 2.2/3.1 With Dysan 8" SSSD Diagnostic Disk $65 check/MO postpaid Chandler Software, 273 W Shore Dr, Marblehead MA 01945 (617) 631-4685

Dr. Dobb's Journal Subscription Problems?

No Problem!
Give us a call and we'll straighten it out. Today.

Outside CA: 800-321-3333
Inside CA: 619-485-9623 or 6947

TECMAR GRAPHICS LIBRARY

TEKMAR lets you do high-res graphics on your TECMAR Graphics Master. Features windowing, viewporting, clipping, axis rotation. Similar to Tektronix graphics. Includes screen dump/restore, Epson screen print, support for HP and Western Graphtec plotters. Includes three curve-fitting programs and graphics application SOURCE CODE. Requires MS-FORT 3.20, or Lahey F77L. Price: $195.
ADVANCED SYSTEMS CONSULTANTS,
18653 Ventura Boulevard, Suite 351,
Tarzana, California 91356
(818) 990-4942

ECLIPSE SYSTEMS

AZTEC C65 + ProDOS
Use Aztec C DOS 3.3 software under ProDOS. Run programs without relinking. System includes recursive Shell, support for pseudo-code, vi like editor with macros, library upgrade and source code. Requires ProDOS user's disk. $49.95.
Eclipse Systems, 223 Matthew Rd., Marion, PA 19066. (215) 667-8354

OWINGS MILLS

OHIO SCIENTIFIC/ISOTRON AND COMPATIBLE COMPUTERS! Users MUST read PEEK (65). Published monthly exclusively for above users. $19/yr U.S. $26/yr Canada; P.O. Box 347, Owings Mills, MD 21117; (301) 363-3268.

BRINGING YOUR PROGRAMS OUT OF THE CLOSET?

DISKETTES & DUPLICATION
Large or Small - Let's Talk!!

5¼" DISKETTES
SS/DD **99¢ EA.** DS/DD **$1.10 EA.**
Minimum order 250 disks.
8" - Color Jackets - 3½" CALL!!

Hundreds of professionals can't be wrong!!!

CALL TODAY!! **800-437-0500**
800-435-7400 In CA
(408) 262-3008

SUPPLY LINE
780 Trimble Rd. • Suite 608
San Jose, CA 95131

NATIONAL PUBLIC DOMAIN

Why reinvent the wheel?
Public Domain Software isn't copyrighted, so no need to pay license fees for thousands of routines; business-utilities-dbase. Complete, latest issue user group disks available to rent 'n copy at your leisure. Send $5.00 for a postpaid directory disk or a SASE to National Public Domain Software, 1533 Avohill Dr., Vista, CA 92083 or (619) 727-1015.

W.B. SOFTWARE DEVELOPMENT

BROWSE for CP/M-68K
for looking at any CP/M-68K file
scroll up, down, left, right
powerful FIND (search) command
100% 68000 assembler—ASCII, HEX
ANSI 3.64 & other terminal types
$49.95 US. Visa, M/C, money order.
WB Software Development,
112-Oakhampton Pl. SW,
Calgary, AB,
Canada T2V 4B2 (403) 238-3216

MICRO COMPUTING SERVICES

CBTREE FOR C PROGRAMMERS
Provides enhanced file handling calls directly into C programs. Maintains balanced B-trees, supports unlimited number of keys and key lengths. Fast, Flexible, Efficient. No royalties. Source Code Incl. $179. MICRO COMPUTING SERVICES 2009 Hileman Road Falls Church, VA 22043 (703) 893-0118

Dr. Dobb's Journal
is pleased to announce the
DDJ Classifieds

Offering low cost advertising reaching thousands of computer programmers and professionals each month. Departments to choose from include:
Software
Hardware
Accessories/Supplies
Career Opportunities
Services
User Groups. Special categories are also available.

RATES: DISPLAY ADVERTISERS: Price per column inch $100. Ad must run in 3 consecutive issues.
LINE ADVERTISERS: Price per line $12 (35 characters per line including spaces). Minimum of 5 lines. 3 consecutive issues. Add $3 per line for boldface type. Add $25 if a background screen is desired.

DISCOUNTS: Frequency discounts for 6 consecutive ads (less 7%) and for 12 consecutive ads (less 15%).

MECHANICAL REQUIREMENTS: Camera ready art or typewritten copy only (phone orders excepted). Display: Specify desired size, include $20 if reduction is required. Line: Specify characters in boldface, caps. Allow 6 weeks for publication.

PAYMENTS: Full payment in advance. Check, money order, Visa, M/C, AmEx.

Run this ad in _____ issues
Size: _____ col x _____ inches or 1 col x _____ lines
Payment by: _____ check _____ m/o _____ Visa _____ M/C _____ AmEx
Card # _____ Exp Date _____
Signature _____
Print Name _____
Company _____
Address _____
City _____ St _____ Zip _____
Phone _____ Current Subscriber _____

Mail to or phone Alex Williams, DDJ Classifieds, 2464 Embarcadero Way, Palo Alto, CA 94303 (415) 424-0600

Listing One

```
trim0:
        call    getprm          ; convert it
        mov     param2,ax       ; save 2nd parameter
        cmp     ax,param1       ; is end column < start column?
        jnb     trim1
        jmp     err3            ; yes, exit with error message

trim1:  mov     count,0         ; starting a new line,
                                ; initialize column counter

trim2:  inc     count           ; count characters
        call    cin             ; read a character
        mov     al,char         ; is it carriage return?
        and     al,07fh         ; (ignore high bit in case
        cmp     al,cr           ;  this is Wordstar file)
        je      trim5           ; yes, found end of line
        mov     ax,count        ; fetch current char count
        cmp     sign,0          ; is this include or exclude call?
        jne     trim4           ; jump, -, exclude range
                                ; proceed, +, include range
        cmp     ax,param1       ; is column counter within
                                ; desired range?
        jb      trim2           ; no, discard this char.
        cmp     ax,param2
        ja      trim2           ; no, discard this char.

trim3:  call    cout            ; yes, use this character
        jmp     trim2           ; get next char.

trim4:  cmp     ax,param1       ; is column counter outside
                                ; of excluded range?
        jb      trim3           ; yes, use this character
        cmp     ax,param2
        ja      trim3           ; yes, use this character
        jmp     trim2           ; no, discard this character

trim5:                          ; found end of line
        call    cout            ; write carriage return
        call    cin             ; read presumed line feed
        call    cout            ; write line feed
        jmp     trim1

trunc:                          ; come here if zero parameter
                                ; to delete trailing blanks from
                                ; all lines.  If - sign was in
                                ; command parameter, also delete
                                ; empty lines completely.

trunc1: mov     count,0         ; initialize column counter
        xor     bp,bp           ; init line pointer

trunc2: call    cin             ; read a character
        mov     al,char         ; is it carriage return
        and     al,07fh         ; (ignore high bit in case
        cmp     al,cr           ;  this is Wordstar file)
        je      trunc3          ; yes, go process end of line
        mov     al,char         ; transfer char. to forming line
        mov     byte ptr ds:[line+bp],al
        inc     bp
        cmp     char,blank      ; is character a space code?
        je      trunc2          ; yes, get next char
        mov     count,bp        ; no, update column count
        jmp     trunc2          ; get next char.

trunc3: xor     bp,bp           ; text string now in LINE
        call    cin             ; discard line feed
        cmp     count,0         ; was line empty?
```

```
        jne     trunc4          ; no, go output it
        cmp     sign,0          ; deleting empty lines?
        jne     trunc1          ; yes, discard this one
        jmp     trunc5          ; no, send cr-lf sequence

trunc4:                         ; now transfer LINE to BUFOUT
        mov     al,byte ptr ds:[bp+line]
        mov     char,al         ; get next char and
        call    cout            ; send it to output
        inc     bp
        cmp     bp,count        ; entire line sent yet?
        jb      trunc4          ; no, send another char

trunc5: mov     char,cr         ; send carriage return
        call    cout
        mov     char,lf         ; and line feed
        call    cout
        jmp     trunc1

exit:   cmp     di,0            ; output buffer empty?
        je      exit1           ; yes
        call    outbuf          ; no, flush it
exit1:  mov     ax,4c00h        ; exit with return code=0
        int     21h             ; if no errors were encountered

err:                            ; print error message and exit.
                                ; DS:DX = addr of message
                                ; CX    = length of message
                                ; AL    = return code
        push    ax              ; save return code
        mov     ah,40h          ; function 40 = write
        mov     bx,stderr       ; handle for error output
        int     21h
        pop     ax              ; retrieve return code
        mov     ah,4ch          ; function 4C = exit
        int     21h

err1:   mov     dx,offset err1msg  ; print "output device error"
        mov     cx,err1len
        mov     al,1            ; return code = 1
        jmp     err

err2:   mov     dx,offset err2msg  ; print "disk is full".
        mov     cx,err2len
        mov     al,2            ; return code = 2
        jmp     err

err3:   mov     dx,offset err3msg  ; print "bad parameter"
        mov     cx,err3len
        mov     al,3            ; return code = 3
        jmp     err

err4:   mov     dx,offset err4msg  ; print "input device error"
        mov     cx,err4len
        mov     al,4            ; return code = 4
        jmp     err

trim    endp

cout    proc    near            ; output contents of "char"
                                ; with autobuffering
        mov     al,char
        mov     byte ptr [di+bufout],al
        inc     di
        cmp     di,buflen       ; buffer full yet?
        jb      cout1
        call    outbuf          ; write buffer
cout1:  ret                     ; back to caller
cout    endp
outbuf  proc    near            ; write buffer to std output
        mov     ah,40h          ; function 40 = write
        mov     bx,stdout       ; predefined handle
        mov     cx,di           ; number of characters
        lea     dx,bufout       ; DS:DX = buffer addr
```

(Continued on next page)

Listing One

```
        int     21h             ; request DOS service
        jc      err1            ; jump, device write error
        cmp     ax,di
        jne     err2            ; jump, disk is full
        xor     di,di           ; initialize output buff pointer
        ret                     ; back to caller
outbuf  endp

cin     proc    near            ; input next char with buffering
        inc     si              ; bump input buffer pointer
        cmp     si,topin        ; buffer exhausted?
        jbe     cin2            ; no, jump
        mov     ah,3fh          ; yes, read some more data
        mov     bx,stdin        ; predefined handle
        mov     cx,buflen       ; max length to read
        lea     dx,bufin        ; DS:DX = input buffer addr
        int     21h             ; request DOS service
        jc      err4            ; jump, input device error
        cmp     ax,0            ; end of file?
        jne     cin1
        jmp     exit            ; yes, goto success exit point
cin1:   dec     ax              ; save offset of top of data
        mov     topin,ax
        xor     si,si           ; zero input buffer pointer
cin2:   mov     al,byte ptr [si+bufin]
        mov     char,al         ; get next char
        ret
cin     endp

getprm  proc    near            ; convert numeric parameter to
                                ; binary and return it in AX
        xor     ax,ax           ; initialize forming answer
        mov     cl,[bx]         ; get first char
        cmp     cl,'-'          ; is it minus sign?
        jne     getp2           ; no, jump
        inc     sign            ; yes, set flag and
getp1:  inc     bx              ; bump command string pointer
                                ; past the "-" sign
        mov     cl,[bx]         ; get next char
getp2:  cmp     cl,'0'          ; at least 1 legal digit?
        jb      getp6           ; no, exit
        cmp     cl,'9'
        ja      getp6           ; no, exit
        jmp     getp4
getp3:  inc     bx              ; advance through string
        mov     cl,[bx]
        cmp     cl,'0'          ; make sure legal digit 0-9
        jb      getp5           ; not digit, jump
        cmp     cl,'9'
        ja      getp5           ; not digit, jump
        mov     dl,10           ; previous answer * 10
        mul     dl
getp4:  sub     cl,'0'          ; add in the new digit
        xor     ch,ch
        add     ax,cx
        cmp     ah,0            ; new answer > 255?
        je      getp3           ; no, keep converting
        jmp     err3            ; yes, illegal parameter, exit
getp5:  cmp     byte ptr[bx],blank ; if not digit, must be blank
        jne     getp6           ; exit, bad parameter
        ret                     ; back to caller
getp6:  jmp     err3            ; ... since too far to reach
getprm  endp                    ; direct with conditional branch

err1msg db      cr,lf
        db      'trim: output device error'
        db      cr,lf
```

```
err1len equ     (this byte)-(offset err1msg)

err2msg db      cr,lf
        db      'trim: disk is full.'
        db      cr,lf
err2len equ     (this byte)-(offset err2msg)

err3msg db      cr,lf
        db      'trim: bad parameter'
        db      cr,lf
err3len equ     (this byte)-(offset err3msg)

err4msg db      cr,lf
        db      'trim: input device error'
        db      cr,lf
err4len equ     (this byte)-(offset err4msg)

bufin   equ     this byte       ; data is read here
                                ; from the standard input

bufout  equ     bufin+buflen    ; data to be written to
                                ; standard output is built here

line    equ     bufout+buflen   ; temporary line buffer

cseg    ends

        end     start
```

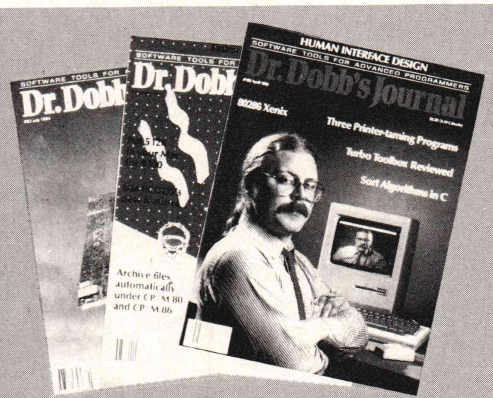
End Listing One

(Listing Two begins on next page)

AVAILABLE

DDJ BACK ISSUES

1982	1983	1984	1985
No. 68—June	No. 76—Feb.	No. 87—Jan.	No. 99—Jan.
No. 69—July	No. 77—March	No. 88—Feb.	No. 100—Feb.
No. 70—Aug.	No. 78—April	No. 90—April	No. 101—March
No. 71—Sept.	No. 80—June	No. 91—May	No. 102—April
No. 72—Oct.	No. 81—July	No. 92—June	No. 103—May
No. 73—Nov.	No. 82—Aug.	No. 94—Aug.	
	No. 83—Sept.	No. 95—Sept.	
	No. 84—Oct.	No. 96—Oct.	
	No. 85—Nov.	No. 97—Nov.	
	No. 86—Dec.	No. 98—Dec.	



TO ORDER: send $3.50 per issue to: Dr. Dobb's Journal, 2464 Embarcadero Way, Palo Alto, CA 94303.

Name

Address

City State Zip

104

Listing Two

```
1                                       name    optest
2                                       page    55,132
3                                       title   'OPTEST -- test MASM bugs'
4                                   ; Demonstrate some anomalies in the Macro Assembler v. 2.
5                                   ; This was run on MASM.EXE file dated 7-18-84
6
7       0000                        cseg    segment para public 'CODE'
8                                       assume  cs:cseg,ds:cseg,es:cseg
9       0100                            org     100h
10                                  ;
11                                  ; SHR and SHL don't distinguish between signed and unsigned data
12      0100  0001                       dw      -2 shr 1
13      0102  0001                       dw      2 shr 1
14      0104  0002                       dw      -1 shl 1
15      0106  0002                       dw      1 shl 1
16                                  ;
17                                  ; EQ and NEQ seem to recognize the decimal and hex equivalents...
18      0108  FFFF                       dw      -1 eq 0ffffh
19      010A  0000                       dw      -1 ne 0ffffh
20                                  ;
21                                  ; but the comparison operators don't ...
22      010C  FFFF                       dw      1 gt -1
23      010E  0000                       dw      1 gt 0ffffh
24      0110  0000                       dw      1 lt -1
25      0112  FFFF                       dw      1 lt 0ffffh
26                                  ;
27                                  ; on MOV instructions, if source operand is missing, no
28                                  ; error message is produced and machine code is generated
29                                  ; as though the source operand were an immediate "0".
30      0114  3D 0000                    cmp     ax,
31
32                                  ; generates the same machine code as
33      0117  3D 0000                    cmp     ax,0
34                                  ;
35                                  ; Assembler generates erroneous machine code when last digit of a
36                                  ; literal is "B" or "D" and the default radix is hexadecimal
37      0010                             .radix  16
38      011A  83 EB 0A                   sub     bx,0a
39      011D  83 EB 00                   sub     bx,0b   ; generates same code as SUB BX,0
40      0120  83 EB 0C                   sub     bx,0c
41      0123  83 EB 00                   sub     bx,0d   ; generates same code as SUB BX,0
42      0126  83 EB 0E                   sub     bx,0e
43
44      0129                        cseg    ends
45                                      end
```

End Listings

A FULL C COMPILER FOR $49.95

YOU WIN!

The Ecosoft Eco-C88 compiler for the 8088 and MSDOS is going to set a new standard for price and performance. Consider the evidence:

Compiler	Eco-C88	Lattice (1)	C86 (1)
Seive	13	11	13
Fib	44	58	46
Deref	13	13	--
Matrix	21	29	27
Price	$49.95	$500.00	$395.00

(1) **Computer Language,** Feb., 1985, pp.73-102. Reprinted by permission.

The Eco-C88 compiler is a full K&R C compiler that supports all data types and operators (except bit fields). Now look at the other features we offer:

★ 8087 co-processor support using a single library. If you install an 8087 later, the software will use it without having to recompile.
★ A robust standard library with over 150 functions, including transcendentals, color, and others.
★ OBJ output for linking with the MSDOS linker (LINK).
★ Error messages in English – no cryptic numbers to look up. A real plus especially if you're just getting started with C.
★ Easy-to-read and complete user's manual.
★ Works with all IBM and compatibles running MSDOS 2.0 (or later).
★ Plus many other features.

For $10.00 more, we will include the source code for the C library functions (excluding transcendentals). For an additional $15.00, we will include our ISAM file handler in OBJ format (as published in the **C Programmer's Library,** Que Publishing). The discount prices for the library source and ISAM only apply at the time the compiler is purchased. Please add $4.00 to cover postage and handling. To order, call or write:

Ecosoft Inc.
6413 N. College Avenue
Indianapolis, IN 46220
(317) 255-6476

MasterCard VISA AMERICAN EXPRESS

ECOSOFT INC.

Eco-C (Ecosoft), MSDOS (Microsoft), UNIX (Bell Labs), CP/M (Digital Research), Z80 (Zilog), 8086, 8087, 8088 (Intel).

Circle **no. 35** on reader service card.

TOTAL CONTROL:

FORTH: FOR Z-80®, 8086, 68000, and IBM® PC

Complies with the New 83-Standard

GRAPHICS • GAMES • COMMUNICATIONS • ROBOTICS
DATA ACQUISITION • PROCESS CONTROL

- ***FORTH*** programs are instantly portable across the four most popular microprocessors.
- ***FORTH*** is interactive and conversational, but 20 times faster than BASIC.
- ***FORTH*** programs are highly structured, modular, easy to maintain.
- ***FORTH*** affords direct control over all interrupts, memory locations, and i/o ports.
- ***FORTH*** allows full access to DOS files and functions.
- ***FORTH*** application programs can be compiled into turnkey COM files and distributed with no license fee.
- ***FORTH*** Cross Compilers are available for ROM'ed or disk based applications on most microprocessors.

Trademarks: IBM, International Business Machines Corp.; CP/M, Digital Research Inc.; PC/Forth + and PC/GEN, Laboratory Microsystems, Inc.

FORTH Application Development Systems include interpreter/compiler with virtual memory management and multi-tasking, assembler, full screen editor, decompiler, utilities and 200 page manual. Standard random access files used for screen storage, extensions provided for access to all operating system functions.

Z-80 FORTH for CP/M® 2.2 or MP/M II, $100.00; **8080 FORTH** for CP/M 2.2 or MP/M II, $100.00; **8086 FORTH** for CP/M-86 or MS-DOS, $100.00; **PC/FORTH** for PC-DOS, CP/M-86, or CCPM, $100.00; **68000 FORTH** for CP/M-68K, $250.00.

FORTH + Systems are 32 bit implementations that allow creation of programs as large as 1 megabyte. The entire memory address space of the 68000 or 8086/88 is supported directly.

PC FORTH + ... $250.00
8086 FORTH + for CP/M-86 or MS-DOS $250.00
68000 FORTH + for CP/M-68K $400.00

Extension Packages available include: software floating point, cross compilers, INTEL 8087 support, AMD 9511 support, advanced color graphics, custom character sets, symbolic debugger, telecommunications, cross reference utility, B-tree file manager. *Write for brochure.*

VISA LMI

Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to ***(213) 306-7412***

MasterCard

Circle **no. 55** on reader service card.

JOHN D. OWENS ASSOCIATES JOHN D. OWENS

LARGE DISCOUNTS FROM JOHN D. OWENS ASSOCIATES

MACROTECH MI-286: 80286/Z-80H DUAL PROCESSOR S-100 CPU BOARD: . . . **$876** MSR RAM: 120NS, high-speed dynamic RAM, Works with CompuPro 8085/8088; MI-286 and others: 256K . . **$556** 512K . . . **$716** STATIC RAM: Substitute for RAM 22 and RAM 23: 256K STATIC: ..... **$960** 512K STATIC: **$1,800**

EMERALD SYSTEMS HARD DISK and TAPE SUBSYSTEMS
High capacity! Up to 280 MB! Emerald has overcome the 32MB DOS limitation! Multiple volumes per physical drive. Back up and restore utilities.

HOUSTON INSTRUMENTS: Plotters: DMP 41 OR 42: . . **$2,397,** DMP 29: . . . **$1,838** DIGITIZERS: DT11 . . **$694;** DT114 . . . . . **$750;** DT11AA . . . . . **$714** New! 14 pen DMP 51 and 52: . . . **$4,796**

ILLUMINATED TECHNOLOGY S-100 COLOR GRAPHICS: **$1,116**

NEC APC III: 80816 MS-DOS system w/ spectacular graphics: . . . . . 20% off list

SEMIDISK 2MB DISK EMULATOR FOR IBM PC and EPSON QX 10: . . . . . **$2,050** S-100 1MB: . . **$1,472** S-100 2MB: . . **$2,091**

LOMAS: 80286, CCP/M and MS-DOS S-100 Systems with COLOR GRAPHICS for IBM-PC compatibility.

RACTER: Interactive conversational software for IBM-PC. Jan 1985 Scientific American called RACTER, ". . . extremely funny." RACTER is light years ahead of Eliza! . . . . . **$69.95.**

MIST + CONEXUS: Integrated software for IBM-PC combining telecommunications, text processing, data base management.

BIG DISCOUNTS on COMPUPRO, IMS, BYAD, many others. Prices & availability subject to change without notice.

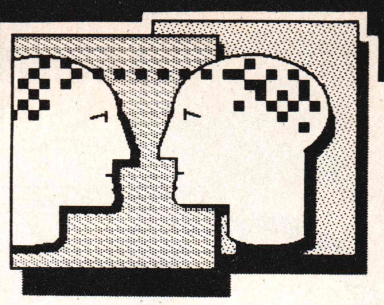
WRITE OR CALL FOR PRODUCT LITERATURE.

WE EXPORT: Overseas Callers: TWX 710 588 2844 (OWENSASSOC NYK). EASY LINK MAILBOX ADDRESS: 62840 768.

ORDER BY PHONE, LETTER, TELEX OR EASY LINK. We accept VISA and Mastercard. Shipping $5 per board in continental USA.

JOHN D. OWENS ASSOCIATES
12 SCHUBERT STREET
STATEN ISLAND, N.Y. 10305
(718) 448 6283 (718) 448 6298
(718) 448 2913

JOHN D. OWENS ASSOCIATES JOHN D. OWENS



by Robert Blum

Alternate Console for SID

Necessity was the mother of invention that spurred Bridger Mitchell of Plu*Perfect Systems to develop this useful modification for ZSID, SID, or DDT. Apparently during the development and debugging of a CRT screen-handling routine, Mitchell realized that he had to find an alternate path for the messages coming from the debugging program. Each time SID displayed a message, it altered the context of the original screen, making it practically impossible to decide whether the screen-handling routine was functioning properly. Even worse, performing an instruction trace completely destroyed the screen contents: as each instruction executed, a new line of information from SID was displayed, scrolling yet another data line off the top of the CRT screen, until the screen was filled with nothing but messages from the debugger. This was no help at all in determining how the routine under test was interacting with the CRT.

Mitchell developed SID2TTY to provide a standard way of interacting with the BDOS to direct application program messages to one display device while providing an additional path for any messages coming from the debugging program. This program dynamically patches the SID debugger to redirect all of its console input and output to the logical TTY: device while allowing the program under test to continue using the CRT: device undisturbed. If you are using either ZSID or DDT, you should use ZSID2TTY and DDT2TTY, respectively, to patch the debugger.

By using a second terminal as the TTY: device to "command" SID, you can debug programs without interfering with standard console input and output. Assuming the usual CON: =CRT: device assignment, a program's standard output will appear on the video monitor exactly as if the debugger were not in use; program input is entered on the standard keyboard. All debugger input comes from the TTY: console, and debugger output appears on the TTY: video monitor. The patch in the Listing (page 107) is only 23 bytes long, the remainder of the code providing the necessary relocation and memory protection services.

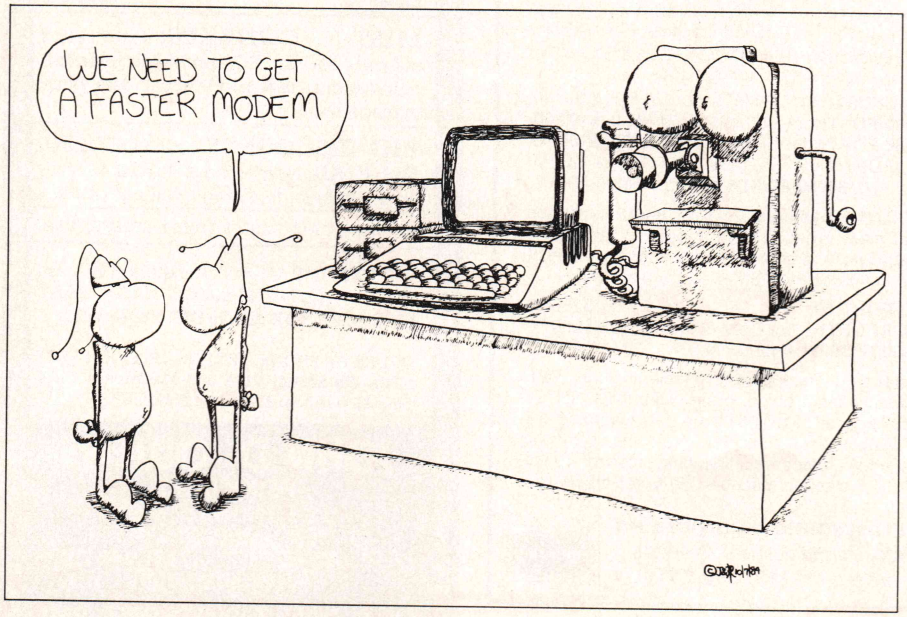
In operation, each time SID calls for BDOS console I/O service, SID2TTY changes the IOBYTE to reflect a console assignment of TTY: and restores it to the original value afterwards. To protect high memory, the patch is relocated below SID and also below any .SYM and .UTL files, which should be loaded first. Operating procedures are well documented within the program source listing.

CP/M Plus Inaccuracy

The CP/M Plus documentation states that the fourth field of a RSX header, labeled PREVIOUS, contains a 16-bit pointer to the previous RSX in the chain or to memory location 5 if no other RSXs are active. This suggests that the PREVIOUS pointer, in fact, always points to the NEXT jump instruction of the previous RSX in the chain. Thus, compatibility is maintained throughout the system, and each RSX is chained in both forward and backward directions.

In actuality, though, this pointer is to location 7, the last byte of the jump NEXT field in the previous RSX, or to memory location 7 if no other RSXs are active. This inaccuracy has no effect on CP/M Plus's operation or on any RSX currently supplied by DRI. But beware if you are planning to write any custom RSXs that might use reverse chaining.

DDJ



Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 198.**

CP/M Exchange Listing

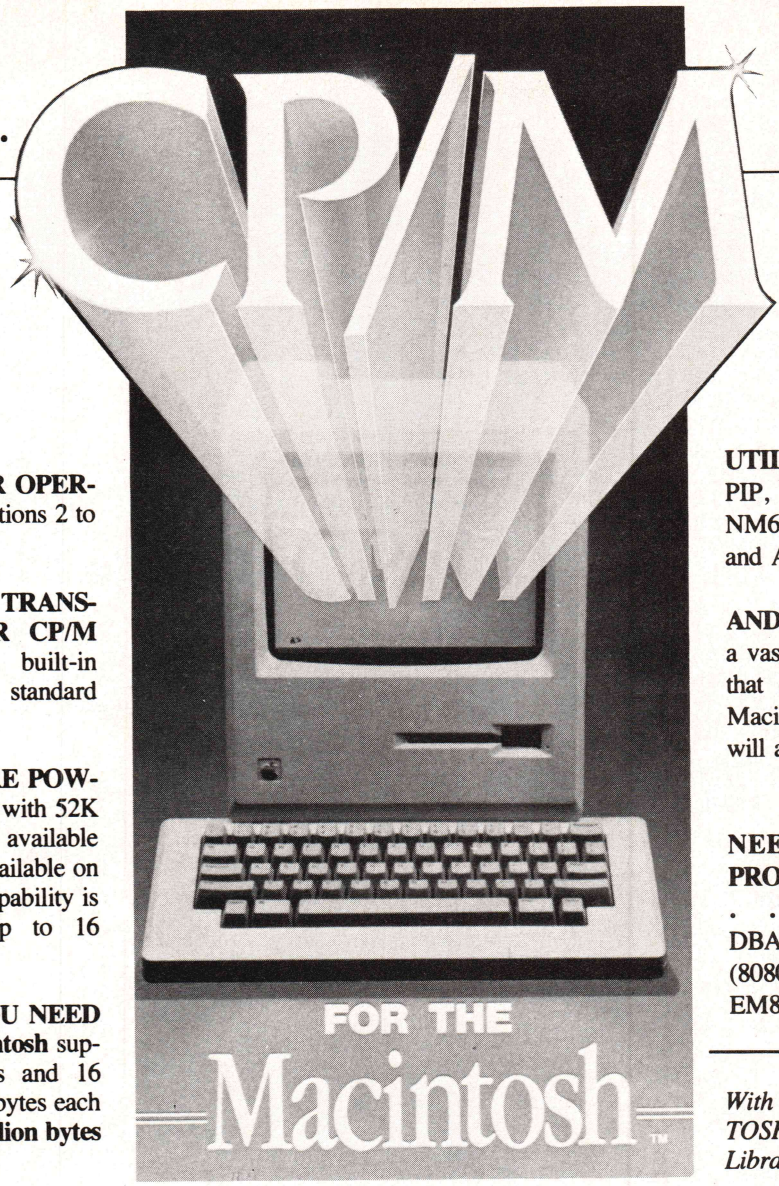
```
;       SID2TTY.ASM v 1.1  01/10/84     B. Mitchell (Plu*Perfect Systems)
;
;
;           A Debugging Console for SID, ZSID and DDT
;
;SID2TTY is a small, public-domain program that
;dynamically patches the SID debugger (Data Resources, Inc.) to
;redirect all of SID's console input and output to the logical TTY:
;device.  (ZSID2TTY and DDT2TTY are the ZSID and DDT versions.)
;
;By using a second terminal as the TTY: device to "command" SID,
;one can debug programs without interfering with standard console
;input and output.  Assuming the usual CON:=CRT: device assignment,
; a program's standard output will appear on the video
;monitor exactly as if the debugger were not in use, and program
;input will be entered on the standard keyboard.  All debugger
;input comes from the TTY: console, and debugger output appears
;on its video monitor.
;
;The patch itself is just 23 bytes -- it changes the IOBYTE each time
;SID calls for BDOS console i/o service and restores it afterwards.
;The patch is relocated to protected high memory, below SID (and
;below any .SYM and .UTL files, which should be loaded first).
;
;Running SID2TTY will display a help message.
;
;Source file:   SID2TTY.ASM for assembly with ASM or MAC.
;               The ZSID and DDT versions are obtained by
;               changing three equates.
;
;Requirements:  CP/M 2.2 - 3.0 with IOBYTE implemented in bios.
;               8080, 8085 or Z80 CPU.
;               SID v. 1.4 or ZSID v. 1.4 or DDT v2.2.
;               External terminal connected to TTY:.
;
;Installation:  Set the equates for SID, ZSID or DDT.
;               If your TTY: device needs initialization or
;               de-initialization, insert the code in place
;               of dummy routines.
;
;Bugs:          Exiting from SID with ^C leaves the console
;               redirected to TTY: (unless the BIOS improperly
;               restores the IOBYTE on warmboots).  The
;               recommended fix is simply to exit with a "G0".
;               Should you happen to hit ^C, reload SID and reset
;               the IOBYTE to its original value, then exit with
;               ^C from the CRT terminal.
;
;Remarks:       1. Although it would be possible to trap the
;               warmboot and restore the IOBYTE, it's difficult to
;               do correctly.  Trapping 0001h violates the basic CP/M
;               addressing convention, making it impossible to
;               run other resident modules that need to locate the BIOS
;               and BDOS.  Trapping the warmboot at the BIOS jump
;               vector does preserve addressing, but it still requires
;               special efforts to maintain compatibility with any higher
;               resident modules, and the trap must remove itself
;               without disturbing them.  Since the application here
;               is manual, interactive debugging, it seems best to
;               favor compatibility and do a manual restoration.
;
;               2. The assembler portion of SID/ZSID can be removed
;               with the "-A" command to gain 680h bytes, but
;               this must be done BEFORE installing the patch
;               with the "g103" command.
;
;Version:       1.0     --      10 October 1984 B. Mitchell
;               1.1     --      10 January 1985 B. Mitchell
;
;Revisions:     Please forward any revisions and improvements
;               to the author.
```

(Continued on next page)

```
;
;Author:         Bridger Mitchell
;               Plu*Perfect Systems
;               Box 1494, Idyllwild CA 92349
;               (714) 659-4432
;
;Note:           Not tested on CP/M 3.0.
;
;
VERS    equ     1$1
;
FALSE   equ     0
TRUE    equ     NOT FALSE
ADDRESS equ     0FFFFH
;
; exactly ONE of the next 3 equates must be TRUE --
;
SID     equ     TRUE
ZSID    equ     FALSE
DDT     equ     FALSE
;
CR      equ     0dh
LF      equ     0ah
BELL    equ     07h
;
IOBYTE  equ     0003h
BDOS    equ     0005h
;
        org     100h
START:  jmp     HELP
        jmp     INSTALL
        jmp     DEINITTTY
;
signature:db    'NEXT  PC'      ;common SID/ZSID/DDT code
siglen  equ     $-signature

        IF SID
;
modsiz  equ     1800h   ;size of debugger module
sigad   equ     0E42h   ;offset to signature
callbd  equ     06A4H   ;offset to call_bdos routine
fn2ad   equ     1004h   ;offset to addr of function 2 bdos call
fn10ad  equ     0FF2H   ; ditto fn 10
fn11ad  equ     1065h   ; ditto fn 11

;
usage:  db      CR,LF,'SID2TTY v. '
        db      vers/10+'0','.',(vers mod 10)+'0'
        db      '  --  Redirect SID console i/o to TTY:',CR,LF
        db      '          -- Usage --',CR,LF
        db      'Load SID, then .SYM and .UTL files',CR,LF
        db      '  #I* filename.SYM',CR,LF
        db      '  #R',CR,LF
        db      '  #ISID2TTY.COM',CR,LF
        db      '  #R',CR,LF
        db      '  #G103       ==>> SID''s console is now TTY:',CR,LF
        db      '  #Ifilename.COM  (or .HEX) to debug',CR,LF
        db      '  #R',CR,LF
        db      '  #D3,3       ==>> display IOBYTE for reference',CR,LF,LF
        db      'To exit from SID, deinitialize the TTY: (if needed) by',CR,LF
        db      '  #ISID2TTY.COM',CR,LF
        db      '  #R0',CR,LF
        db      '  #G106',CR,LF
        db      'Then use "G0" (not "^C") to exit from SID.',CR,LF
        db      '(This message can be displayed under SID by the command:'
        db      CR,LF,'   G100  or   C100 before loading ''filename'')'
        db      CR,LF,LF,'$'
;
errmsg: db      CR,LF,BELL,'Can''t find SID v 1.4!$'
        ENDIF   ;IF SID

        IF ZSID
```

(Continued on page 110)

The Macintosh Software Library just got fatter . . .



It's about TIME. AND SPEED.

FASTER COMPUTER OPERATION with disk operations 2 to 5 times faster.

QUICK PROGRAM TRANSFER FROM OTHER CP/M COMPUTERS using built-in transfer utilities and standard CP/M file structure.

RUNS BIGGER MORE POWERFUL PROGRAMS with 52K transient program area available on 128K Mac, 300K available on 512K system and the capability is built-in to manage up to 16 Megabytes of RAM.

PLUG-IN WHAT YOU NEED ...CP/M for the Macintosh supports up to 2 printers and 16 Drives of 512,000,000 bytes each for a maximum of **8 Billion bytes of storage.**

MORE DISK SPACE AVAILABLE ... each disk drive has 360 bytes available, often eliminating the need for additional drives.

CHOOSE ANY PRINTER letter quality or dot matrix ... any serial printer will work just fine.

MEDIA PROBLEMS?...NONE ... all that is needed to transfer programs from 5¼ to 3½ is a modem transfer program to operate with our Modem 7 Communications Compatible Program ... this enables programs to be transferred from **XEROX, OSBORNE, HEATHKIT, KAYPRO, RADIO SHACK** and others.

UNLIMITED EXPANDABILITY is made possible with the total and easy control of serial expansion ports which allows many forms of plug-ins.

UTILITIES ... such as STAT, PIP, DDT, ED, LINK68, INIT, NM68, SIZE, LO68, RELOC, and AS68.

AND We are currently compiling a vast Library of CP/M software that can be ported over to the Macintosh and our Macrolibrary will also be available soon.

NEED POPULAR WORD PROCESSING PROGRAMS? . . . . Run WORDSTAR® DBASEII® AND other CP/M (8080) 2.2 Programs with our EM80 Emulator . . **ONLY $195.!**

With CP/M FOR THE MACINTOSH the Macintosh Software Library just got fatter!

THE 10 FEATURES:

1. 6 Disk System
2. Documentation from Digital Research and I.Q.
3. MacroAssembler ***(no extra charge)***
4. Modem 7 Compatible Communications Transfer Program ***(no extra charge)***
5. C Compiler ***(no extra charge)***
6. Text Editor ***(no extra charge)***
7. Menu Program (puts menus on your programs) ***(no extra charge)***
8. Standard Printer Driver ***(no extra charge)***
9. Copy Program ***(no extra charge)***
10. Leir-Seigler ADM3A Terminal Emulation Program ***(no extra charge)***

All This For Only $395 Retail

Call or write for your Macintosh CP/M Software Catalog.

I.Q. Software
2229 E. Loop 820 N.
Ft. Worth, Texas 76118
(817) 589-2000

Macintosh is a registered trademark of Apple Computer, Inc. CP/M is a registered trademark of Digital Research, Inc. WORDSTAR is a registered trademark of Micropro. DBASE II is a registered trademark of Ashton-Tate. ©1985 I.Q. Software

CP/M Exchange Listing (Listing continued, text begins on page 106)

```
;
modsiz  equ     2200h
sigad   equ     1682h
callbd  equ     0EAFH
fn2ad   equ     1880h
fn10ad  equ     186Eh
fn11ad  equ     18E1h
;
usage:  db      CR,LF,LF,'ZSID2TTY v. '
        db      vers/10+'0','.',(vers mod 10)+'0'
        db      '  --  Redirect ZSID console i/o to TTY:',CR,LF,LF
        db      '          -- Usage --',CR,LF
        db      'Load ZSID, then .SYM and .UTL files',CR,LF
        db      '  #I* filename.SYM',CR,LF
        db      '  #R',CR,LF
        db      '  #IZSID2TTY.COM',CR,LF
        db      '  #R',CR,LF
        db      '  #G103       ==>> ZSID''s console is now TTY:',CR,LF
        db      '  #Ifilename.COM  (or .HEX) to debug',CR,LF
        db      '  #R',CR,LF
        db      '  #D3,3       ==>> display IOBYTE for reference',CR,LF,LF
        db      'To exit from ZSID, deinitialize the TTY: (if needed) by',CR,LF
        db      '  #IZSID2TTY.COM',CR,LF
        db      '  #R0',CR,LF
        db      '  #G106',CR,LF
        db      '(This message can be displayed under ZSID by the command:'
        db      CR,LF,'   G100  or   C100 before loading ''filename'')'
        db      CR,LF,LF,'$'
;
errmsg: db      CR,LF,BELL,'Can''t find ZSID v 1.4!$'
        ENDIF   ;IF ZSID

        IF DDT
;
modsiz  equ     1000h
sigad   equ     0A71h
callbd  equ     06A2h
fn2ad   equ     0BCEh
fn10ad  equ     0C25h
fn11ad  equ     0BBCh
;
usage:  db      CR,LF,LF,'DDT2TTY v. '
        db      vers/10+'0','.',(vers mod 10)+'0'
        db      '  --  Redirect DDT console i/o to TTY:',CR,LF,LF
        db      '          -- Usage --',CR,LF
        db      'A>DDT DDT2TTY.COM',CR,LF
        db      '  -G103       ==>> DDT''s console is now TTY:',CR,LF
        db      '  -Ifilename.COM  (or .HEX) to debug',CR,LF
        db      '  -R',CR,LF
        db      '  -D3,3       ==>> display IOBYTE for reference',CR,LF,LF
        db      'To exit from DDT, deinitialize the TTY: (if needed) by',CR,LF
        db      '  -IDDT2TTY.COM',CR,LF
        db      '  -R0',CR,LF
        db      '  -G106',CR,LF
        db      '(This message can be displayed under DDT by the command:'
        db      CR,LF,'   G100 before loading ''filename'')'
        db      CR,LF,LF,'$'
;
errmsg:db       CR,LF,BELL,'Can''t find DDT v 2.2!$'
        ENDIF   ;IF DDT
;
;
; Display usage message & quit.
;   This is executed if SID2TTY is run as a .COM file
;   It can also be C'd or G'd from the debugger.
;
HELP:   lxi     d,usage
print:  mvi     c,9
        call    bdos
```

(Continued on page 112)

Free Catalog!

Your 80-page guide to computer supplies and accessories–including complete new product descriptions.

- Packed with over 1600 products for microcomputers, minicomputers, and word processors – many available nowhere else.
- Big special section devoted to new supplies and accessories.
- Comprehensive product descriptions – including more than 475 full-color photos – clearly explain features and benefits.
- Easy-to-use cross reference guides to magnetic media, ribbons, and more–along with the industry's most complete cable guide.
- Helpful suggestions and tips, ranging from flexible disk care to proper ribbon selection to useful application ideas.

Phone toll-free 1-800-547-5444
In California, call 1-800-547-5447.

inmac™

Phone toll-free 1-800-547-5444 or send coupon today.*

Inmac Catalog Dept.
2465 Augustine Drive
Santa Clara, CA 95054

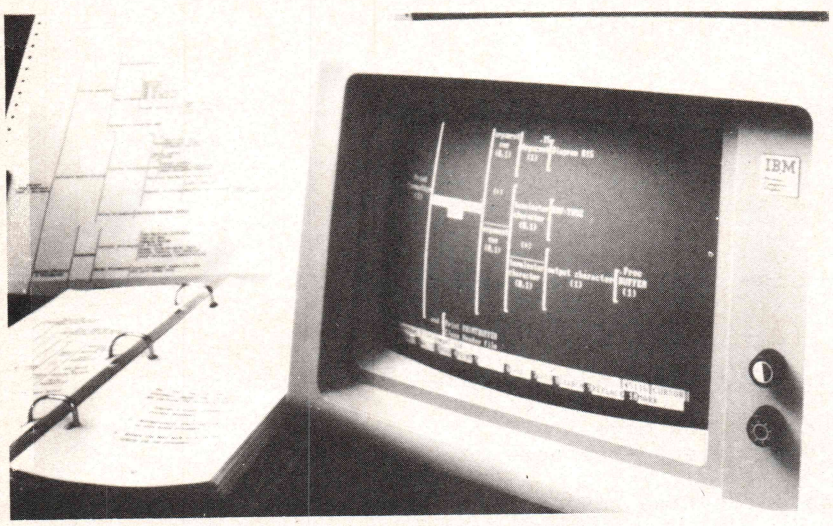
Please rush my free copy of the Inmac Catalog. I understand there is no obligation whatsoever.

NAME _____
COMPANY _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____ PHONE _____

*In California, call 1-800-547-5447.

Circle **no. 53** on reader service card.

WARNIER/ORR
Structured Diagramming Tool



SDT: Structured Diagramming Tool

Features: Full screen editing • Automatic formatting including brackets • Menu Driven • Block moves • Cursor, Search, Label navigation • Procedures • Global & Selective Replace •Begin/End blocks •Detail hiding • Diagram families & more. Tutor & Manual.

Requires: IBM PC, DOS 2.x, 256K RAM, 5¼" Diskette, Mono Display.

Price: $325.00
Demo: $ 20.00

To order SDT call with credit card number or mail check or company P.O.

Varatek Software Architects
523 Winter St., No. Andover, Ma. 01845
(617) 685-7003

MasterCard VISA

Circle **no. 60** on reader service card.

YOU NEED A GOOD LIBRARY

POWER PACKS™

COMPLETE SOURCES NO ROYALTIES

COMPREHENSIVE C Power Packs include over 1000 functions which provide an integrated environment for developing your applications efficiently. "This is a beautifully documented, incredibly comprehensive set of C Function Libraries."
– Dr. Dobb's Journal, July 1984

USEFUL "...can be used as an excellent learning tool for beginning C Programmers..."
– PC User's Group of Colorado, Jan. 1985

FLEXIBLE Most Compilers and all Memory Models supported.

RECOMMENDED "I have no hesitation in recommending it to any programmer interested in producing more applications code, using more of the PC capabilities, in much less time." — Microsystems, Oct. 1984

- **PACK 1: Building Blocks I** $149
  DOS, Keyboard, File, Printer, Video, Async
- **PACK 2: Database** $399
  B-Tree, Virtual Memory, Lists, Variable Records
- **PACK 3: Communications** $149
  Smartmodem™, Xon/Xoff, X-Modem, Modem-7
- **PACK 4: Building Blocks II** $149
  Dates, Textwindows, Menus, Data Compression, Graphics
- **PACK 5: Mathematics I** $99
  Log, Trig, Random, Std Deviation
- **PACK 6: Utilities I (EXE files)** $99
  Arc, Diff, Replace, Scan, Wipe

Master Card/Visa, $7 Shipping, Mass. Sales Tax 5%

ASK FOR FREE DEMO DISKETTE

NOVUM ORGANUM INC.

SOFTWARE HORIZONS INC.

165 Bedford St., Burlington, MA 01803
(617) 273-4711

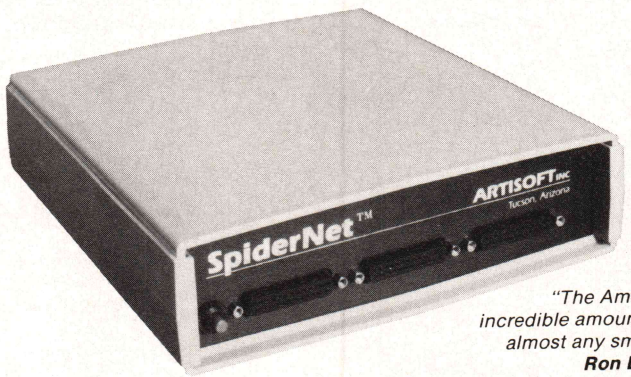
Circle **no. 90** on reader service card.

```
        lxi     h,0             ;is debugger in use on default stack?
        dad     sp
        mov     a,h
        cpi     2
        rnc                     ;return to caller (ccp)
        rst     7               ;return to debugger
;
ERR:    lxi     d,errmsg
        jmp     print
;
;
; Search for debugger signature in high memory.
;   Search downward to find first image of the debugger,
;   in case other copies are lying around.  This also
;   allows other resident modules to exist above the
;   debugger.  Debugger is always located on a page boundary.
;
INSTALL:
        lhld    1
        lxi     d,-(modsiz+3)
        dad     d               ;start at max possible addr
search: push    h
        lxi     d,sigad
        dad     d
        call    compar
        pop     h
        jz      found
        mov     a,h
        cpi     10h             ;quit looking at 1000h
        jc      err
        dcr     h               ;try 1 page lower
        jmp     search

compar: lxi     d,signature
        mvi     b,siglen
complp: ldax    d
        cmp     m
        rnz
        inx     h
        inx     d
        dcr     b
        jnz     complp
        ret
;
; Debugger located.  Calculate run-time addresses & patch
;   into code and debugger.
;
found:  push    h               ; save base addr of debugger
        lxi     d,callbd        ;put 3 run-time addresses into code
        dad     d
        xchg
        lxi     h,patch1        ;1: the call_bdos routine address
        mov     m,e
        inx     h
        mov     m,d
        lhld    6               ;2: the protect address/base of high code
        shld    patch2
        lxi     d,-codelen      ;protect the code
        dad     d               ;(don't trace from here to lpend)
        shld    6
        push    h               ;save run-time base of high code
        lxi     d,saviob-code
        dad     d
        shld    patch3          ;3: the save-iobyte address

        pop     d               ;get destination
        push    d
        lxi     h,code          ;move the code into place
        mvi     b,codelen
```

(Continued on page 114)

FINALLY!

Communications Hardware and Software that's Inexpensive, Easy to Use and Guaranteed.



"The Amazing SpiderNet . . . an incredible amount of utility available to almost any small-sized application."
Ron Exner, 12/84 Hardcopy

SNARE™—SpiderNet's Callback Security System — $595

- Protects ANY dialup computer system from unauthorized intruders
- Authorized users are called back at specific locations
- SNARE secures three modem lines at less than $200/line
- Up to three separate computers can use SNARE simultaneously
- Uses Hayes-compatible modems or others like DEC's DF03
- Stores 70 authorized users, expands to 150 (300+ soon)

SpiderNet—Printer/Plotter Sharing Unit — $495

- Shares expensive RS-232 plotters, daisy wheel or laser printers
- Makes laser printers affordable for your office or department
- Two to five micros or minis can share a peripheral
- Connect and use—no software modifications required!
- Optional 64K buffer for spooling

SpiderNet—Computer and Peripheral Networking — $495

- Six port, intelligent software-controlled RS-232 switch
- Share multiple peripherals between systems
- Interconnect computers to share data and files
- Three pairs of ports can be connected simultaneously
- Links ports at different baud rates
- Programmable: perfect for custom RS-232 control application

SpiderNet—Multiplexer, Terminal Concentrator and more — $495

- Five to one RS-232 multiplexer/demultiplexer
- 5×1 or 4×2 concentrator for expanding terminal ports

ENVOY™—Telecommunications Software — $49.95

- Access to electronic mail, remote systems and data networks
- Error free, text and binary file transfers via XMODEM or ANSI X3.28
- Smart terminal mode with capture buffer, autodial and autologin
- Easy to use, menu driven, compact and high speed
- Utilities menu for copy, type, print, erase and rename files
- For IBM PC, PCjr, PC compatibles, Sanyo MBC-55X, CP/M-80 or -86

30-day money-back guarantee on all products

ARTISOFT INC
Box 41436, Tucson, Arizona 85717
(602) 327-4305

Circle **no. 7** on reader service card.

CP/M Exchange Listing

(Listing continued, text begins on page 106)

```
lp:     mov     a,m
        stax    d
        inx     h
        inx     d
        dcr     b
        jnz     lp
;
lpend:  pop     b               ;patch debugger's console io calls
        inx     b
        inx     b
        inx     b               ; ..to point at code entry ('divert')
        pop     d               ;de = base of debugger
        lxi     h,fn10ad        ;index into debugger module
        dad     d
        mov     m,c             ;plug 3 diversion addresses into calls
        inx     h
        mov     m,b
        lxi     h,fn2ad
        dad     d
        mov     m,c
        inx     h
        mov     m,b
        lxi     h,fn11ad
        dad     d
        mov     m,c
        inx     h
        mov     m,b
        jmp     INITTTY         ;do any user-supplied initialization
;
;
; The REDIRECTION CODE, which is moved up just below the debugger
;
patch2  equ     $+1
CODE:   jmp     ADDRESS         ;jmp to debugger entry
divert: lxi     h,iobyte
        mov     a,m
patch3  equ     $+1
        sta     ADDRESS         ;save the iobyte
        ani     0fch            ; assign CON:  to TTY: for debugger
        mov     m,a
patch1  equ     $+1
        call    ADDRESS         ;debugger's call_bdos routine
saviob  equ     $+1
        mvi     a,00            ;restore user's iobyte
        sta     iobyte
        mov     a,l             ;get bdos return param back to A
        ret
codelen equ $-code

;
; User-supplied TTY: initialization routine.
;   Use this to set baud rate, channel parameters, etc.
;
INITTTY:
        rst 7                   ;return to debugger on TTY: device
;
; User-supplied TTY: de-initialization routine.
;   Use this to restore TTY: device to regular settings.
;
DEINITTTY:
        rst 7                   ;return to debugger on TTY: device

        END
```

End Listing

Introducing the MIX Editor

(with Split Screen - both horizontal and vertical)

A Powerful Addition To Any Programmer's Tool Box

Full Screen Editing
WordStar Key Layout
Custom Key Layouts
Terminal Configuration
Help Files
Backup Files

Introductory Offer Only

29.95

30 Day Money Back Guarantee

Programmable
Macro Commands
Custom Setup Files
Mnemonic Command Mode
Multiple File Editing
Split Screen Editing

For PCDOS/MSDOS (2.0 and above/128K) • IBM PC/Compatibles, PC Jr., Tandy 1000/1200/2000, & others
For CPM80 2.2/3.0 (Z80 required/64K) • 8" SSSD, Kaypro 2/4, Osborne I SD/DD, Apple II, & others

Great For All Languages

A general purpose text processor, the MIX Editor is packed with features that make it useful with any language. It has auto indent for structured languages like Pascal or C. It has automatic line numbering for BASIC (255 character lines). It even has fill and justify for English.

Terminal Configuration

A utility for defining terminal features (smart features included) allows the editor to work with any terminal. Over 30 of the most popular terminals are built-in.

Custom Key Layouts

Commands are mapped to keys just like WordStar. If you don't like the WordStar layout, simply change it. Any key can be mapped to any command. You can also define a key to generate a string of characters, great for entering keywords.

Split Screen

You can split the screen horizontally or vertically and edit two files simultaneously.

Macro Commands

The MIX Editor allows a sequence of commands to be executed with a single keystroke. You can define a complete editing operation and perform it at the touch of a key.

Custom Setup Files

Custom keyboard layouts and macro commands can be saved in setup files. You can create a different setup file for each language you use. The editor automatically configures itself using a setup file.

Command Mode

Command mode allows any editor command to be executed by name. It is much easier to remember a command name versus a complicated key sequence. Command mode makes it easy to master the full capability of the editor. Frequently used commands can be mapped to keys. Infrequent commands can be executed by name.

Editor Commands

The editor contains more than 100 commands. With so many commands, you might think it would be difficult to use. Not so, it is actually extremely simple to use. With command mode, the power is there if you need it, but it doesn't get in your way if you don't. Following is a list of some of the commands.

Cursor Commands

Left/Right/Up/Down
Tab Right/Tab Left
Forward Word/Backward Word
Beginning of Line/End of Line
Scroll Up/Scroll Down
Window Up/Window Down
Scroll Left/Scroll Right
Top of File/Bottom of File
• • •

Block Commands

Copy/Move/Delete
Read/Write
Lower Case/Upper Case
Fill/Justify
Print

File Commands

Directory (with wild cards)
Show File/Help File
Input/Output File
Delete File/Save File

Other Commands

Split Screen/Other Window
Find String/Replace String
Replace Global/Query Replace
Delete Line/Undelete Line
Delete Word/Undelete Word
Insert Mode/Overwrite Mode
Open Line/Join Line
Duplicate Line/Center Line
Set Tab/Clear Tab
• • •

To Order: **Call Toll Free 1-800-622-4070,** (Illinois only **1-800-942-7317**)
Mix Editor ___ $29.95 + shipping ($5 USA/$10 Foreign) Texas residents add 6% sales tax

Visa ___ MasterCard ___ Card # _____ Exp. Date _____
COD ___ Check ___ Money Order ___ Disk Format _____
Computer _____ Operating System: MSDOS ___ PCDOS ___ CPM80 ___
Name _____
Street _____
City/State/Zip _____
Country _____
Phone _____

MIX software
2116 E. Arapaho
Suite 363
Richardson, Tx 75081
Dealer Inquiries Welcome
Call (214) 783-6001

MIX software
2116 E. Arapaho
Suite 363
Richardson, Tx 75081
(214) 783-6001

MSDOS is a trademark of Microsoft
PCDOS is a trademark of IBM
CPM80 is a trademark of Digital Research
WordStar is a trademark of MicroPro

For everyone who ever tried doing five things at once

The perfect computer program for someone as busy as you. It lets you keep several other programs working at once.

Do you ever go in so many directions so fast not even a computer can keep up with you?

Well, now an IBM Personal Computer can—thanks to IBM TopView.

TopView is a new kind of software that lets you switch between other programs as quickly as you can change your mind, even run several programs at the same time.

Once you load TopView into your computer, you load the other programs you use most—as many as your computer's memory will permit.

After that, the greatest distance between two programs is just a couple of keystrokes, or (optional) mouse moves.

There's no waiting and a lot less diskette swapping.

But when you're *really* busy is when TopView really shines, letting you do many jobs simultaneously.

For example, you can print a letter, while you search a file, while you analyze a spreadsheet, while your clock/calendar reminds you that your automatic dialer is about to place a call for you.

Little Tramp character licensed by Bubbles Inc., s.a.

...IBM presents TopView.

And you can see everything through on-screen "windows" and control it all with easy-to-use pop-up menus.

You can even make unrelated programs work together; say a "Brand Y" spreadsheet with a "Brand Z" word processor.

But simplest of all is a certain "Brand IBM", namely the IBM Assistant Series—for filing, writing, planning, reporting and graphing.

Many other popular programs also work with TopView, and the number is growing.

Naturally, the more computer memory you have, the more TopView can help you. At least 512K is recommended.

And the price is only $149*.

Beyond that, all you need is to be the kind of person who never does a single thing all day, but who wants to do everything, at once.

To learn more, call an IBM marketing representative, or visit an IBM Product Center or Authorized IBM PC or Software Dealer.

For the store nearest you, and a free brochure, call 800-447-4700. (In Alaska and Hawaii, 800-447-0890.)

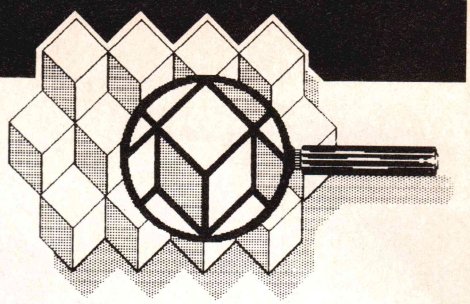
IBM®

Personal Computer Software

*IBM Product Center price.

Circle **no. 68** on reader service card.

OF INTEREST



by Alex Ragen

If you've ever traveled outside the United States, you've probably noticed that the household electricity varies from country to country. The line voltage may be 110 or 120 or 220 or 240 volts, and the frequency may be either 50 or 60 Hz. Even if your electric shaver or hair dryer has a little switch somewhere to flip back and forth among all the possible combinations, you still may find that you can't use it because the plug doesn't fit the wall receptacle. It turns out that almost every country in the world has its own unique plug shape, which is not only different but also superior to every other country's plug shape. Does this remind you of anything?

Software and hardware expend a tremendous amount of computing energy in converting between languages, protocols, disk formats, operating systems, and so on. You can run CP/M 80 or MSDOS on your Macintosh or use your IBM PC (and compatibles, of course!) to write Apple II disks or run a 68000 board. Whatever designers intended users to do on one configuration, they have managed to implement on another as well. This Babel of standards and not-so-standards is chaotic, confusing, pointless, and counterproductive—and we've all had lots of fun and worked hard getting around it.

Perhaps one day we'll live in a perfect world. Everything will interface with everything else, Macintosh add-ons won't be called Mac-this or Mac-that, and a word processor will emerge that is so good that nobody will want to write another one.

C Language

Lattice Inc. is offering an upgrade kit to users of Microsoft's MSDOS C compiler. For $150 and the original Microsoft diskettes, Lattice will provide version 2.20 of its C compiler, the C-SPRITE debugger, and a typeset manual. The regular retail price of the package is $675, and the offer is good until July 31, 1985. Contact Lattice Inc. at P.O. Box 3148, Glen Ellyn, IL 60138 (312) 858-7950. **Reader Service No. 101.**

QTools, a programmer's toolbox of 19 tools adapted from Unix to PCDOS, is available for $49.95 postpaid from QCAD Systems, 1164 Hyde Ave., San Jose, CA 95129 (800) 538-9787 or (408) 255-5574 in California. **Reader Service No. 103.**

A new C language **scientific subroutine library** that provides IBM PC users with 112 pretested and precompiled mathematical and statistical subroutines is available from Wiley Professional Software. The subroutines include the most commonly used operations such as differentiation, polynomials, probability, numerical integration, regression, differential equations and others. Over 400 pages of documentation provide a description of methodology, notes on special considerations, source code, test programs, and results. The library requires the Lattice C compiler, version 2.12 or later, and a two-drive IBM PC. Contact Leslie Bixel, Wiley Professional Software, 605 Third Ave., New York, NY 10158 (212) 850-6788. **Reader Service No. 105.**

TGL Inc. has released version II of **The Converter**, an IBM PC program that automatically converts UCSD, MT+, and MS-Pascal source programs into equivalent C programs. Ted Lewis of TGL points out that The Converter correctly handles nested procedures, separately compiled units or modules, and intrinsic functions; it also enables developers to move quickly from UCSD or MSDOS to Unix. Contact Mr. Lewis at TGL Inc., 4400 Sulphur Springs Rd., Corvallis, OR 97330 (503) 745-7476. **Reader Service No. 107.**

Thunder Software has announced version 2.0 of its **Thunder C**, the only C compiler available for the Apple Pascal and Apple ProDOS environments. Thunder C generates native 6502 assembly language routines, and its developers rate the new version as 300 percent faster than the old one. It runs under Apple Pascal 1.1 or 1.2, although the ProDOS version requires an external ProDOS macro assembler. Price is $49 plus $3 shipping. The company also offers LINKIT, a linking loader and library generator; XREF, a Pascal cross-reference utility; and ASSYST, a 6502 assembler. Contact Thunder Software at P.O. Box 31501, Houston, TX 77231 (713) 728-5501. **Reader Service No. 109.**

Hippo-C is the only (so far) C compiler for the Macintosh with a source-level debugger—or so its developers claim. Level 1 provides access to over 380 Toolbox and Quickdraw routines and features a compiler, editor, standard C library, on-line tutorial, debugger, linker, shell command processor, many sample programs, and over 200 pages of documentation. It's priced at $149.95. Level 2 includes all Level 1 features plus an optimizing compiler, assembler, and full floating-point support. It sells for $399.95. Contact Hippopotamus Software at 1250 Oakmead Pkwy., Sunnyvale, CA 94086 (408) 738-1200. **Reader Service No. 111.**

Modula 2

Release 1.10 of **Modula-2/86 for PCDOS** is now available from Logitech. A library of programming tools is also available. Contact Chris Cale at Logitech, 805 Veterans Blvd.,

Redwood City, CA 94063 (415) 365-9852. **Reader Service No. 113.**

Modula-2 for Z80-based CP/M systems is available from Hochstrasser Computing AG, Leonhardshalde 21, CH-8001 Zurich, Switzerland, phone 01/47 55 48. The cost is sFr 400 or about $150. **Reader Service No. 115.**

MSDOS

Guess who is throwing its hat into the cutthroat world of the IBM PC add-on market? None other than Intel, which has established a **Personal Computer Enhancement Operation.** The first two products are its well-known 8087 and 80287 math co-processors (what's next, the 8086?). Contact Intel at 520N.E. Elam Young Pkwy., Hillsboro, OR 97124-6497 (503) 629-7369. **Reader Service No. 117.**

PC-68K is an upgrade package—plug-in board and software—that adds development support for Motorola's 68000 family to IBM's PC XT and PC AT running PCDOS. PC-68K provides a symbolic debugger, linker/locator, Motorola compatible macro assembler, and IEEE floating-point package. Pascal and C compilers and communications utilities are extra-cost options. The board features an 8 MHz 68000 cpu and 256 – 1024K of RAM. Prices start at $2995 for the 256K version. Contact Carrie Ann Moran at Language Resources, 4885 Riverbend Rd., Boulder, CO 80301 (303) 449-8087. **Reader Service No. 119.**

The **Buddy System** enables a support person located at a central facility to "take over" a remotely located IBM PC in order to resolve a customer's problem. The developer has been using it internally for three years to support its products and now offers it to others at $199. You load the Buddy System when you turn on your PC; thereafter, it's dormant until you activate the "copilot" mode and contact the support center via modem. The support person then observes you, the user, as the application is executed and can even take control of the application to show you what to do. A number of other features, such as help, screen snapshot, screen copy to file, and record/replay support session, are designed to increase the productivity of the support operation. Contact Edward Murphy at Solvation Inc., 302 Turnpike Rd., Southboro, MA 01722 (617) 481-9390. **Reader Service No.121.**

Phoenix Software Associates has introduced an IBM PC/AT-compatible ROM BIOS that was developed under strict controls specifically to ensure originality and to avoid copyright infringement suits. The company had already developed versions for the PC and PC/XT. The PC/AT version is offered to OEMs for unlimited use licensing at a price of $440,000. The PC and PC/XT versions are $290,000. Contact Richard Levandov at 1420 Providence Hwy., Suite 101, Norwood, MA 02062 (617) 769-7020. **Reader Service No. 123.**

Ryan McFarland's **RM/Cobol** is now available for the IBM PC/IX user. It was already available for the IBM PC under PCDOS, as well as for the IBM Series 1 under Unix and the IBM 370 computers under VM/CMS. The new RM/Cobol is priced at $230 for the runtime version and $750 for

SOFTERM PC™

MICRO-TO-MAINFRAME
CONCURRENT COMMUNICATIONS
EXACT TERMINAL EMULATIONS

SOFTERM PC CONNECTS YOU WITH MAINFRAMES.
Now you can access information services, bulletin boards, electronic mail systems, and even your company's computer from home or office. Softerm PC includes keyboard macros, a built-in phone book for automatic dialing, simultaneous capture to print or disk, and much more. Softerm PC's extensive file transfer capabilities provide automatic execution using command files, support for popular protocols like XMODEM, and our own Softrans™ protocol which comes with an adaptable program for your host computer.

INTEGRATES COMMUNICATIONS WITH ANY PROGRAM.
Now you can add concurrent communications to all your favorite programs. Just load Softerm PC into your computer's memory so it will remain resident while you use other programs. Then, with a single keystroke, you can interrupt the current program and use Softerm to print or display files, dial a host computer, use terminal mode, or initiate a file transfer. You can return to the interrupted program and continue working while the other tasks are performed concurrently.

MAKES YOUR IBM® PC WORK EXACTLY LIKE ANY MAJOR TERMINAL.
Softerm PC includes exact emulations of more than 30 popular terminals. All keyboard and display functions are provided for both conversational and block modes.

SOFTERM PC IS A COMMUNICATIONS BARGAIN.
Softerm PC is only $195. Plus, you get an unconditional 30-day money-back guarantee. So ask your dealer for a demonstration today.

TERMINAL EMULATIONS
TTY Compatible
ADDS Regent 20, 25, 40, 60
ADDS Viewpoint, Viewpoint/78
Data General D200, D410
Datapoint 3601, 8220*
Digital Equipment VT52, VT102, VT220*
Hazeltine 1400, 1410, 1500, 1520
Hewlett Packard 2622A
Honeywell VIP 7205, 7801, 7803, 7813*
IBM 3101 Model 10 and 20
Lear Siegler ADM-2, ADM-3A, ADM-5
TeleVideo 910, 921, 925, 950
Texas Instruments Model 931*
TRS-80 Model 16 Console
User Defined
*Coming Soon

Softerm is also available in versions for the Apple II and Tandy Model 1000 and 2000 personal computers.

IBM is a registered trademark of IBM.

SOFTRONICS
3639 New Getwell, Suite 10, Memphis, TN 38118
For orders and information call 901-683-6850

Circle **no. 49** on reader service card.

ConIX™

NOW ONLY $79.95!

If you think you're missing out on innovative software developments because nobody is writing for CP/M™-80, take a look at us. We've adapted UNIX™ features to CP/M like never before, and with the kind of professional, quality-controlled product that you deserve. That product is none other than the critically acclaimed ConIX Operating System.

ConIX can provide any 48K+ CP/M-80 or compatible system with I/O Redirection and Pipes (uses memory or disk), perfected User Areas, Command and Overlay Path Searching, Auto Screen Paging, 8Mb Print Buffering, 22 new SysCalls, Function Keys, "Virtual" disk system, Archiver (saves over 50% disk), extensive command language, 300+ variables, 100+ commands, pull-down menu, and much more! Uses as little as 1/2K RAM! Runs with CP/M for true data and software compatibility. Installs easily without any system mods!

The ConIX package lists at $165 and has been advertised and sold internationally to many enthusiastic customers since October 1983. As a special limited offer, we've lowered the price of the complete ConIX system by 50% to only $79.95! Don't miss this opportunity to bring your 8-bit micro back into the software revolution. Order your copy of ConIX today!

Price includes manual, 8" disk, and user support. 5¼" conversions available. Contact your local dealer, or buy direct and add shipping: $4.50 UPS, $10 Canada, $25 overseas. NY residents add sales tax.

Computer Helper Industries Inc.
P.O. Box 680 Parkchester Station, NY 10462
Tel. (212) 652-1786 (for information/orders)

"We're helping your computer work better for you!"

UNIX: AT&T Bell Labs, CP/M: Digital Research, ConIX: Computer Helper Ind.

Circle **no. 22** on reader service card.

FINALLY...

THE C JOURNAL

The C Journal will help YOU use C on YOUR machine —IBM PC™, CP/M™, Macintosh™, or UNIX™-based — micro, mini, or mainframe.

It's the ONE publication for programmers, software managers, and other computer professionals who need to keep aware of developments in the industry's fastest-growing language.

- regular columns for novice through advanced C programmers
- software product and book reviews
- tips on working with major compilers and operating systems
- news from the ANSI standards committee and the industry

For **FREE** sample issue and discount subscription information, write, call, or circle our reader service number. **The C Journal** is a quarterly publication, and costs $28/year (add $9 for overseas airmail).

Subscriptions/Advertising:
Christina Gardner
(201) 989-0570

Editorial:
Rex Jaeschke
(703) 860-0091

another independent publication from

InfoPro Systems
3108 Route 10
P.O. Box 849
Denville, NJ 07834

MasterCard VISA

Circle **no. 91** on reader service card.

GREAT BARGAINS IN HISTORY:

The Louisiana Purchase	$15,000,000	1803
Seward's Icebox (Alaska)	$ 7,200,000	1867
The Island of Manhattan	60 Guilders	1626
The EC Text Editor	**$49.50**	**1985**

The men who made these historical purchases could perceive a real value — like the person who buys EC. Need a well-crafted programmers editor? Then take a look at what EC has to offer. . .and if you really need to spend over $100 to have confidence in a text editor, let us know and we'll raise the price.

- **DOS INTERFACE — MORE THAN A BRIDGE TO DOS!**

Automate compiling, Linking, testing - do it all with one keystroke! Capture error messages, then look at them as you edit your program.

Scroll through previous DOS output - even if it has gone off the screen. Rewrite your last DOS command with one keystroke. Convenient!

Run any program inside EC - any DOS command, compiler, DB manager - even BASIC!

- **COMMAND AND TEXT MACROS.**
- **EC IS EASY TO LEARN -** List is Alt-L, Find is Alt-F, Undo is Alt-U, Jump is Alt-J, . . . get the picture?

FULL-FEATURED DEMO. Price: just shipping cost. Its only limitation - the 8k file size!

- **EDIT MULTIPLE FILES IN WINDOWS**
- **FILE SIZE LIMITED ONLY BY AVAILABLE MEMORY**
- **ON-LINE CALCULATOR -** Bases: hex, octal, binary, decimal & ASCII. Bitwise shift, AND, OR, XOR. Integer arithmetic.
- **ADDITIONAL FEATURES —** Color, extended ASCII support, auto-indent, ignore case, auto-tab for C.
- **INTEGRAL SPOOLER AND PRINT FORMATTER.**

SOURCE

C SOURCE
12801 Frost Road
Kansas City, Mo 64138

(816) 353-8808

*** **30 Day Money-Back Guarantee** ***

Requires 192k, DOS 2.0+, PC compatible. COD, MC, VISA. Shipping charge - $5. Dealer, OEM inquiries invited.

Circle **no. 19** on reader service card.

the full version. Contact Ryan McFarland Corp., 609 Deep Valley Dr., Rolling Hill Estates, CA 90274 (213) 541-4828. **Reader Service No. 125.**

A full ANSI **Fortran 77 compiler** for the IBM PC has been announced by Lahey Computer Systems. The complete package, including a 250 page manual, customer telephone support, and newsletters, costs $477 and requires 256K RAM and an 8087 coprocessor. The compiler claims Lattice C compatibility. Contact the vendor at 31244 Palos Verdes Dr. West, Suite 243, Rancho Palos Verdes, CA 90274 (213) 541-1200. **Reader Service No. 127.**

Scroll and Recall lets the IBM PC user scroll back through 27 pages of previously displayed screens while saving you DOS commands, which you can then recall, edit, and reuse without having to type them in again. The cost is $69. Contact Dennis P. Olenick, Opt-Tech Data Processing, P.O. Box 2167, Humble, TX 77347 (713) 454-7428. **Reader Service No. 129.**

The **GTP Program Development System** is a productivity tool for Turbo Pascal programmers. Basically, it generates the Turbo Pascal statements for handling screen data entry. It requires PCDOS or MSDOS version 2.0 or higher, and lists for $99.95. Contact AEF Software, P.O. Box 928, Katy, TX 77449 (713) 391-8570. **Reader Service No. 131.**

FREECOPY is functionally equivalent to DISKCOPY, which is distributed as part of PCDOS. Although it is basically DISKCOPY reverse engineered, the author claims not to have violated the proprietary rights of either IBM or Microsoft. Commented source code is included. The author is placing the program in the public domain but does request a $25 contribution. Contact Donald Buresh, Squire Buresh Associates, 18 Dorothy Rd., Millbury, MA 01527 (617) 865-3435. **Reader Service No. 133.**

Modular Bridge is a product that allows you to transfer files between PCDOS and the PICK PC-XT operating system. Contact Modular Software, P.O. Box 204, Union City, GA 30291 (404) 964-7171. **Reader Service No. 135.**

HFORMAT and **HTEST** are two programs that address the problems of formatting hard disks on the PC family. The programs test and perform the equivalent of factory formatting on the drives. Contact Marcus Kolod at Kolod Research, P.O. Box 68, Glenview, IL 60025 (312) 291-1586. **Reader Service No. 137.**

A **Universal PROM programmer** for the IBM PC is offered for $250 by Advanced Microcomputer Systems of 6802 N.W. 20th Ave., Ft. Lauderdale, FL 33309 (305) 975-9515. **Reader Service No. 139.**

HelpDOS is a menu-driven on-screen reference for PCDOS and includes a technical dictionary. It requires PCDOS or MSDOS, version 2.0 or later, and is available for $49.95 from Help Technologies, P.O. Box 50834, Palo Alto, CA 94303 (415) 856-3431. **Reader Service No. 141.**

Superkey, a resident keyboard enhancer for data encryption and macro processing, has been announced by Borland International, the Turbo

Thinking of... 8087, High-Speed Math Support?

THINK C-86 OPTIMIZING C86 FROM COMPUTER INNOVATIONS

The 8087/80287 Math Co-processor chips provide a major leap in the computing capability of the IBM family of personal computers. The fast, accurate, math computational ability provided by these chips not only speeds typical applications but opens a range of applications once reserved exclusively for mini and mainframe systems. The key question is HOW CAN THE C PROGRAMMER TAKE ADVANTAGE OF THE 8087/80287's capabilities? The answer is C86 by Computer Innovations.

EXECUTION SPEED, ACCURACY, IN-LINE CODE

C86 has offered the highest level of 8087/80287 support **since the chips became available**. Specialized libraries make program development a snap. A math function library provides full trig, log and a wide variety of other functions.

But most important, C86 produces **IN-LINE** code. This means that math functions can be performed by the co-processor in immediate succession WITHOUT separate subroutine calls. This eliminates library call "overhead" and results in the highest execution speed attainable.

LIBRARY SOURCE PROVIDES CONFIDENCE - CONTROL

Library Source Code is INCLUDED. - It always has been with C86. This means you have total control over your development environment, and total control means better and faster code.

TECH SUPPORT - GET INTO 8087/80287 PAINLESSLY

Computer Innovations' experienced tech support team is available to get you through the rough spots and provide the assistance/advice/expertise required to get you into 8087/80287 quickly and efficiently. Plus you can take advantage of Computer Innovations' user newsletter, technical notes, and On-Line bulletin board. No other Compiler offers more.

CALL FOR FREE BULLETIN

To receive Computer Innovations' free bulletin entitled "How C86 Takes Advantage of Intel 8087/80827 Math Co-processors," or for more information about C86 Call:

800-922-0169

C86 for all PC-MSDOS computers. $395.00. Visa, Mastercard, personal check and corporate P.O.'s accepted.

COMPUTER INNOVATIONS, INC.
980 Shrewsbury Avenue,
Tinton Falls, NJ 07724 (201) 542-5920

Circle **no. 96** on reader service card.

Pascal people. The program is basically a shortcut to routine entry of often-used commands, like those required for logging onto RBBS services or formatting letters and spreadsheets. The user can bring up Superkey, move around in the program, and exit with single-keystroke operations, as well as pull its full-screen macro editor down on top of the main program. It also provides on-line help, automatic recall of the last 20 entered commands, cut and paste capability, data encryption/decryption, and screen burn-in protection. Contact Borland International, 4113 Scotts Valley Dr., Scotts Valley, CA 95066 (408) 438-8400. **Reader Service No. 143.**

SEE is an editor for "professional programmers" from Prologic Corp., 31324 Via Colinas, Suite 111, Westlake Village, CA 91362 (818) 991-5062. **Reader Service No.145.**

A new version of **PC/Forth** for the PC/AT, running under 80286 XENIX 3.0, has been announced by Laboratory Microsystems, 3007 Washington Blvd., Suite 230, P.O. Box 10430, Marina del Rey, CA 90295 (213) 306-7412. **Reader Service No. 147.**

The Echo PC2 is a speech synthesizer card for the IBM PC. The manufacturer, having previously provided similar products for the Apple, claims to be the first to produce one for the IBM PC. It features two speech modes: a limited vocabulary (about 700 words) in a natural-sounding female voice and an unlimited vocabulary in a robotic-sounding voice. Some 400 English language and pronunciation rules have been utilized to produce the computer-synthesized speech. Software can control pitch and volume. The board fits into the PC's short slots and comes with a speaker, software, and an instruction manual. The price is $149.95. Contact Street Electronics Inc., 1140 Mark Ave., Carpinteria, CA 93013 (805) 684-4593. **Reader Service No. 149.**

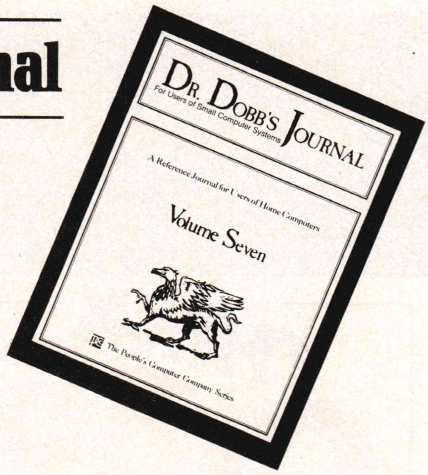
BIS7705 allows an IBM PC to emulate all the functions of a Honeywell 7700/7705 terminal. The price is $695, and it's available from IE Systems Inc., 112 Main St., Newmarket, NH 03857 (603) 659-5891. **Reader Service No. 151.**

CP/M 80

DSD80 is a fully DDT-compatible full-screen symbolic debugger for 8080, 8085, and Z80-based CP/M 80 systems. DSD80 includes port I/O, string searching, and symbol definition. The Z80 instruction set is fully supported using either extended Intel or Zilog mnemonics. There is an on-line help facility, and the program comes with a 50-page user's manual. The price is $125. Contact Soft Advances, P.O. Box 49473, Austin, TX 78765 (512) 478-4763. **Reader Service No. 153.**

CP/M users needn't feel left out in the cold by the latest fad of pop-up programs. Poor Person Software has announced **Write-Hand-Man**, which includes a notepad, phone book, desk calendar, file and directory viewing programs, and a communication program. Users familiar with CP/M programming can add new functions. Write-Hand-Man requires CP/M 2.2

Dr. Dobb's Journal



Bound Volumes

Every Issue Available For Your Personal Reference.

We are pleased to offer this special discounted price to **DDJ** readers who order directly from us. From the nostalgia of Volume One—with authors like Steve Wozniak, Dennis Allison, Sol Libes, and more—to the technical maturity of Volume Seven, **Dr. Dobb's Journal Bound Volumes** are the ideal addition to your reference collection. They contain many issues which are no longer available and you get twelve issues for the price of seven individual back issues!

Send $26.75 for volume 1, $27.75 each for volumes 2–6, $30.75 for volume 7, or $165 for all seven and SAVE!

Please add the following per book: $2.50 for UPS, $1.25 for U.S. Mail, or $3.25 for Foreign surface mail. Foreign Airmail rates available on request. Delivery times are one week for UPS or 6–10 weeks for U.S. or Foreign Mail.

Mail To: Dr. Dobb's Journal
2464 Embarcadero Way, Palo Alto, CA 94303

104

Will Your PC Handle a Volume Larger Than 32 MB?

280 MBytes in one volume

Emerald Systems has broken the 32 MByte DOS barrier and shattered popular attitudes about the performance of micro computers. Our disk drives will easily handle a single logical volume up to 280 MBytes in size (unformatted) on the IBM-PC, XT, AT* and compatibles. Speed? Emerald drives aren't just fast, they're *FAST*—no more trips to the coffee pot while files load.

Emerald delivers the "extras"

The extras means support of the most popular Local Area Networks, password access, support of the VENIX, XENIX, QNX, PC/IX and DOS 3.x operating systems, and user configured, main-frame style, disk caching. Plus, you're not limited to one volume—use up to 24, if you need them, even on one physical disk. You can even spread one volume across two drives. You can have up to 240 MBytes of formatted storage inboard your PC. Or, you can insert an Emerald sub-system, with integral backup and six PC expansion slots, right under your monitor, without increasing your PC's "footprint" even one inch.

Backup protection

Backup of large files is mandatory. Emerald can supply you with ¼ inch, 60 MByte, cartridge or ½ inch, reel-to-reel, backup systems. The ½ inch system can even translate your mainframe (EBCDIC) files to ASCII.

When operated under Emerald's Backup and Restore Utility (BRU) software, both backup files automatically, allow streamer or file oriented operation and file by file access of data. And, they do it fast—our ¼ inch tape drive operates at 5.4 MBytes per minute. That means you can backup 60 MBytes, on one cartridge, in 11 minutes.

Cost Effective

Emerald products translate directly into dollars and "sense." Why pay mainframe prices to do a job your PC can accomplish for a fraction of the cost?

Still not convinced?

Why don't you pick up the phone? Dial (619) 270-1994, and ask to speak with an applications engineer. You can find out about our large installed customer base, learn about some of the applications environments our systems are performing in, or just request literature. We'll even give you references in your area. Find out what your PC can do with a logical volume bigger than 32 MBytes.

Emerald Systems Corporation
Mainframe Storage for Micros
4757 Morena Blvd.
San Diego, CA 92117
Telex: 323458 EMERSYS
Easylink: 62853804

Emerald
Systems Corporation

**IBM-PC, XT & AT™ IBM Corp.*
Mainframe Storage for Micros, Backup and Restore Utility, BRU, & Emerald ™ Emerald Systems Corp.

Circle **no. 114** on reader service card.

Replace 19 floppies with 1 diskette?

TeamMate does it!

Now your Apple IIe can have high capacity data storage like a Winchester while you still enjoy the convenience and familiarity of a standard floppy.

TeamMate's flexible disk drive from **Eastman Kodak** gives you 2.78 megabytes of data storage on one rugged diskette. You get unlimited expandable storage and access to Winchester capabilities...**at a very low cost.**

Installation takes only minutes, thanks to TeamMate's software transparency with ProDOS. No software drivers or modifications are needed.

Here are the TeamMate subsystems available now:

TM 2203...A single Kodak flexible disk drive giving you 2.78MB on one diskette.
TM 2206...Two Kodak drives for a total of 5.56MB of online, expandable storage.
TM 2210...10MB of Winchester storage, with the option of later adding a Kodak flexible drive.
TM 2213...A 10MB Winchester combined with the Kodak 2.78MB flexible drive for built-in backup.
TM 2232...32MB of Winchester storage—the ultimate in high performance mass storage.

TeamMate Model	Quantity 1-2	Quantity 3-9	Quantity 10+
TM 2203	$ 895	$ 761	$ 716
TM 2206	$1295	$1101	$1036
TM 2210	$1295	$1101	$1036
TM 2213	$1695	$1441	$1356
TM 2232	$2195	$1866	$1756
Kodak Diskettes	$ 15	$ 14	$ 12

All TeamMate subsystems include drive(s), controller, TeamMate chassis, power supply, necessary cabling, and manual. Kodak subsystems include a Kodak diskette.

Team• Mate

Yes, I want to expand data storage on my Apple IIe. Please send me the following:

Price includes UPS surface shipping.
Allow 3-4 weeks delivery.

Quantity:

_____ TeamMate 2203 @ _____ ea.
_____ TeamMate 2206 @ _____ ea.
_____ TeamMate 2210 @ _____ ea.
_____ TeamMate 2213 @ _____ ea.
_____ TeamMate 2232 @ _____ ea.
_____ Kodak diskettes @ _____ ea.

Subtotal _____
CA & MA residents, add sales tax _____
For optional UPS Blue Label delivery add $10 per unit _____
Total Enclosed $ _____

Ship To:

Name _____
Address _____
City _____
State _____ Zip _____
Daytime Phone _____

Payment Enclosed:
☐ check ☐ money order
☐ cashier's check

Make check payable to:
Data Technology Corporation
Dept. TM-1A
2525 Walsh Avenue
Santa Clara, CA 95051

For fastest delivery, call:
In California, call collect:
408-986-9545

1-800-824-3724

Circle **no. 67** on reader service card.

and will run on all CP/M machines. The price is $49.95. The program is available from Poor Person Software, 3721 Starr King Circle, Palo Alto, CA 94306 (415) 493-3735. **Reader Service No. 155.**

MB+ Tools is a set of programmer productivity tools for Pascal MT+. It's currently available for CP/M 80 and CP/M 86, and it should be available for MSDOS late this year. The price is $175. Contact Michael Nunamaker, Minnow Bear Computers, P.O. Box 2233, Champaign, IL 61820-8233 (217) 398-6883. **Reader Service No. 157.**

Macintosh

MacCharlie is not a trampburger but a $985 box that attaches to the Macintosh and allows it to run programs written for the IBM PC. MacCharlie also allows Macintosh users to connect to IBM PC serial networks and to use IBM PC-compatible printers. Contact Robert Barrett at Dayna Communications, 50 South Main St., Suite 530, Salt Lake City, Utah 84144 (801) 531-0600. **Reader Service No. 153.**

Mighty Mac is a personal information manager priced at $99 from Advanced Logic Systems at 1195 E. Arques Ave., Sunnyvale, CA 94086 (408) 730-0307. **Reader Service No. 155.**

FileMaker (the third product in the MACWARE line) is a single-file data base system that allows a user to create a variety of custom-designed reports and forms that include text and graphics. FileMaker takes full advantage of the "visually active" Macintosh environment and functions within the Macintosh Office recently announced by Apple. Contact Forethought Inc., 1973 Landings Dr., Mountain View, CA 94043 (800) MACWARE. **Reader Service No. 129.**

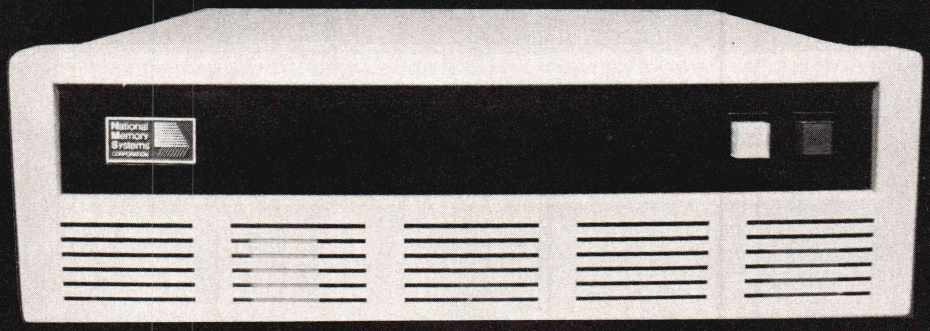
DDJ

Reader Ballot
Vote for your favorite feature/article.
Circle Reader Service **No. 201.**

Try a mainframe tradition on your microcomputer today...

Desk Top Capacities

1000 mb
700 mb
335 mb
170 mb
85 mb



Rack Mount Capacities

1000 mb
960 mb
480 mb

The NMS PC 8000 Series
- 8-inch technology-based storage
- Access times of 15 milliseconds

...National Memory Systems provides

Desk top storage to 700 megabytes:
With enhanced features for 1985, the NMS PC 8000 stays ahead of any herd with compact 8-inch technology Winchester Disk Storage systems ... used in mainframe and minicomputer installations. NMS originated use of these techniques in micro applications more than 18 months ago. We set the standard for high-capacity, high-performance mass storage systems.

Compatibility:
IBM PC-AT, XT, standard PC, and all compatibles, as well as a version to run your Texas Instruments PC, too! Ask about our multifunction, single slot disk tape controller.

Ideal for networking and distributed processing:
You gain up to 5 times the speed of 5¼-inch technology disk systems. NMS Disk Systems will not impose limitations on your computer.

File-by-File Tape Back-up:
Add our stand-alone PC.25 60 mb File-by-File and streamer ¼-inch tape system, or ½-inch PC 9000 Series, ANSI-IBM format compatible 9-track system, and you have the total solution.

Laser Optical Disk:
Be revolutionary with the 1 gigabyte NMS PC 007, a non-erasable archival system that gives you 1000 megabytes of on-line data at 1/20th the cost of tape media.

Easy Upgrades:
Mix disk capacities, disk with tape, disk with Laser Disks to meet your needs. One single-slot controller does it all.

Call Us:
All products provided by NMS are protected with our standard 1-year warranty, and optional Nationwide field services program.

National Memory Systems CORPORATION

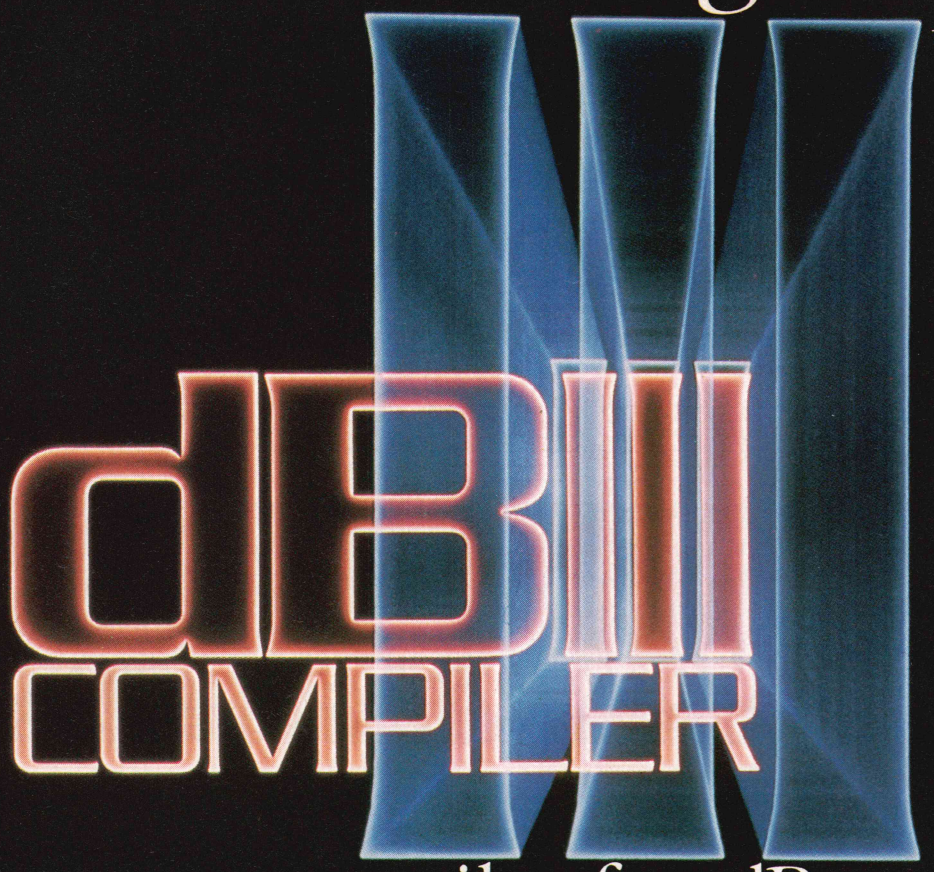
355 Earhart Way
Livermore, CA 94550
(415) 443-1669

Phone 415-443-1669

TWX 9103866006
TELEX 821892NMSUD

Circle **no. 44** on reader service card.

No waiting!



The new compiler for dBase III®

EASE OF USE

dBIIICompiled code can use both dBASE II and dBase III data files, eliminating the need to convert from one format to another. And dBIIICompiler's unique design lets you concentrate on your dBASE program rather than complex memory management problems.

INDEPENDENCE

Buy dBIIICompiler once, and compile and distribute as many applications as necessary at no additional cost. WordTech imposes no license fees, and compiled applications will execute without dBASE III or RunTime.®

PORTABILITY

Versions exist for MS-DOS, as well as for the IBM PC. Cross-environment linkers between versions enable you to run your dBASE III programs on most MS-DOS machines.

SPEED

Compiled applications run faster than under the dBASE III interpreter, and are more compact.

SECURITY

Compilation is the best way of protecting your program source code.

Suggested retail price: $750; Cross-environment linkers: $100 Special introductory offer (expires July 31, 1985); dBIIICompiler and one cross linker: $750

Site licenses available

WORDTECH SYSTEMS

Tools for Thinkers since 1982

Circle no. 51 on reader service card.

WORDTECH SYSTEMS, INC. P.O. Box 1747 Orinda, CA 94563 TX 503599 (415) 254-0900

dBASE III, dBASE II, dBASE, RunTime™, Ashton-Tate, Inc. MS-DOS® Microsoft, Inc. IBM PC® International Business Machines, Inc.

PERFORMANCE ACCELERATORS FOR CP/M-80 & MP/M-80

WSOPTION for WORDSTAR 3.0 & 3.3 (installs itself right into WordStar)

FAST

Speeds up action of WordStar functions so you spend less time waiting for WordStar to take action on your commands.

PRODUCTIVE

A superior print-while-edit capability is included. You can now use one terminal or micro to edit one file while printing another without reducing your typing speed!

CONVENIENT

At print time you can select 1 of 4 printers under CP/M or 1 of 8 printers under MP/M. Under MP/M you do a proper attach and detach of the printer so it is always free when you are not using it. You are informed if the printer is in use at print time so you do not hang up. Many additional features installed via menu.

MPMPLUS for MP/M-80

A group of programs and modules designed to improve your console and disk response under MP/M-80. A performance increase of 2 to 3 times is usual for disk I/O.

Send $35.00 for each item plus $2.00 postage.

INCLUDE DISK FORMAT REQUIRED

Continuum Microsystems Ltd.
21 McCarty Crescent
Markham, Ontario
Canada L3P 4R4

Use your Visa or M.C.

(416) 294-8536

WordStar reg. MicroPro, CP/M MP/M reg. D.R.I.

Circle **no. 29** on reader service card.

C Users' Group

Over 45 volumes of public domain software including:

- compilers
- editors
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details

The C Users' Group

415 E. Euclid • Box 97A
McPherson, KS 67460
(316) 241-1065

Circle **no. 17** on reader service card.

PROGRAM PACKAGING DISKS, SUPPLIES

- IBM style cloth binders and slips for 5½ x 8½. Imprinting available in small quantities.
- Vinyl binders for 5½ x 8½ and 8½ x 11 with clear pockets on front and spine for labeling.
- Vinyl packaging in several styles.
- Floppy disk pages for 5¼ and 3½ disks.
- Continuous paper for IBM size binders with 3 holes punched. 2500 sheets for only $40.00.
- 5¼ DS/DD bulk disks—$149/100. No labels or envelopes. Blank & printed envelopes & labels available.
- Low prices on packaged disks—BASF, KODAK, Elephant, Memorex. Disks for IBM AT and Mac in stock.
- *Many other products. Call or write for FREE CATALOG. Fast service.*

Anthropomorphic Systems, Limited

Department DDJ
376 E. St. Charles Road
Lombard, IL 60148
(312) 629-5160

Circle **no. 3** on reader service card.

RGB COLOR BUSTER ''' RGB

CONVERT THE RGB OUTPUT OF YOUR COLOR BOARD FOR THE IBM PC OR AN IBM PC COMPATIBLE TO COMPOSITE B/W WITH 16 LEVELS OF GRAY '

RUN ALL APPLICATIONS REQUIRING A COLOR MONITOR ON YOUR COMPOSITE BLACK AND WHITE MONOCHROME DISPLAY '

WITH THE ABILITY TO ELIMINATE OR HIGHLIGHT ANY COLOR GROUP, THE COLOR BUSTER OUTPERFORMS ANY OTHER COLOR TO B/W CONVERTER AVAILABLE.

BUY YOURS TODAY AT THIS SPECIAL INTRODUCTORY PRICE FOR A LIMITED TIME ONLY '

ONLY $69.95
(Calif. residents add 6.5% tax)

STS ENTERPRISES
5469 ARLENE WAY
LIVERMORE, CA 94550
(415) 455-5358

Circle **no. 47** on reader service card.

No source code for your REL files?

REL/MAC

converts a REL file in the Microsoft™ M80 format to a ZILOG™ or 8080 source code MAC file with insertion of all public and external symbols.

- **REL/MOD** lists library modules
- **REL/VUE** displays the bit stream
- 50 page manual with examples
- free brochure available
- **REL/PAK** includes all of the above

REL/PAK for 8080 only $99.95
REL/PAK for Z80 & 8080 ..... $134.95
on 8"SSSD disk for CP/M™ 2.2

Send check, VISA, MC or C.O.D. to

MICROSMITH
COMPUTER TECHNOLOGY
P.O. BOX 1473 ELKHART, IN 46515
1-800-622-4070
(Illinois only 1-800-942-7317)

Circle **no. 41** on reader service card.

NEW!

Advanced Trace86™

Symbolic Debugger & Assembler Combo

- Full-screen trace with single stepping; Even backstepping!
- Write & Edit COM & EXE programs
- Conditional breakpoints (programmable)
- Switch between trace and output screen; Or set up two monitors
- 8087, 80186, 80286, 80287 support
- Write labels & comments on code
- Polish hex/decimal calculator
- and more . . . Priced at $175.00

To order or request more information contact:

Morgan Computing Co., Inc.

2520 Tarpley Rd. Suite 500
(214) 245-4763 Carrollton, TX 75006

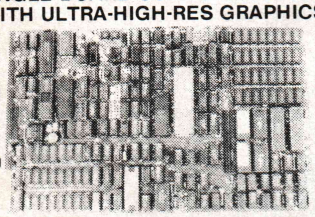
Circle **no. 128** on reader service card.

FOR THE **BEST** OF US . . .

THE CYPHER™

A COMPLETE 68000 & Z80A SINGLE BOARD COMPUTER SYSTEM WITH ULTRA-HIGH-RES GRAPHICS!!

FREE 68000 FORTH AND CYPHER-DOS (CPM-80 COMPATIBLE AND 68000 BASIC)



FREE 68000 FORTH AND CYPHER-DOS (CPM-80 COMPATIBLE AND 68000 BASIC)

- 68000 & Z80A DUAL PROCESSORS (BEST OF BOTH WORLDS' OPTIONAL Z80 HI)
- 256K to 1 MEGABYTE MEMORY. (4164 OR 41256 DRAM)
- DOUBLE DENSITY FLOPPY DISK CONTROLLER (8", 5¼" OR 3½", WD 2793)
- DMA CONTROLLER FOR FAST IMAGE TRANSFERS TO/FROM VIDEO MEMORY. (INT 8237)
- 2 RS232 SERIAL PORTS (ZSIO)
- 24 BIT ADDRESS MANAGEMENT FOR Z80
- 4 LAYER P.C.B. (9"x 14½")
- RUNS CP/M-80 2.2 CP/M-80 3.0 CP/M-68K CYPHER DOS, RAM DISK, 68000 BASIC IN ROM. NEC 7220 TERMINAL EMULATION.
- ULTRA HIGH RESOLUTION GRAPHICS, 128K. PROGRAMMABLE UP TO 1024 x 1024 RESOLUTION (NEC 7220. GREAT FOR CAD SYSTEMS!)
- REAL TIME CLOCK (MULTITASKING CAPABILITY!)
- TWO CHANNELS OF D/A AND A/D, 12 BIT RESOLUTION (MUSIC/ ROBOTICS/ LAB WORK!)
- 16K TO 64K MONITOR EPROM
- 4K TO 64K STATIC RAM
- PROGRAMMABLE BAUD RATE GENERATOR
- PARALLEL ASCII KEYBOARD INPUT.
- FULL 68000 EXPANSION BUS (90 PIN, HEADER BUFFERED)

LOWER PRICES! NOW 1 MEGABYTE CYPHER AVAILABLE.

- MANUAL $20.00
- BARE BOARD, EPROMS, Z80 BIOS, 68000 BIOS, Z80 MONITOR, 68000 MONITOR AND UTILITIES $399.95
- COMPLETE "CYPHER" WITH **256 K DRAM** 128K VIDEO DRAM, NEC 7220, REAL TIME CLOCK, A/D, D/A DISC CONTROLLER, SERIAL I/O, ASSEMBLED AND TESTED $1249.95
  PARTS KIT $1149.95
  EXPANDED MEMORY OPTIONS - INQUIRE
- COMPLETE "CYPHER" WITH **1 MEGABYTE DRAM**, 128K VIDEO DRAM, NEC 7220, REAL TIME CLOCK, A/D, D/A DISC CONTROLLER, SERIAL I/O, ASSEMBLED AND TESTED $1399.95
  PARTS KIT $1299.95
- KEYBOARD $94.95
- SWITCHING POWER SUPPLY $154.95
- CASE $94.95
- HARD DISC INTERFACE PLUG-IN CARD $150.00

SHIPPING CHARGES
ALL PRICES ARE IN U.S. DOLLARS
SHIPPING IN- SHIPPING EXCESS WILL BE REFUNDED
PRICES SUBJECT TO CHANGE WITHOUT NOTICE
CP/M IS A TRADEMARK OF DIGITAL RESEARCH

VISA master charge

MOTOROLA INTEL

MOTEL COMPUTERS LIMITED
174 BETTY ANN DRIVE, WILLOWDALE,
TORONTO, ONTARIO, CANADA M2N 1X6
(416) 221-2340

Circle **no. 39** on reader service card.

ICs PROMPT DELIVERY!!!

SAME DAY SHIPPING (USUALLY)

OUTSIDE OKLAHOMA: NO SALES TAX

INCREASE XT MOTHERBOARD TO 640 Kbyte CONSUME NO EXPANSION SLOTS! $10.980

DYNAMIC RAM			
256K	256Kx1	150 ns	$ 4.99
128K	128Kx1	150 ns	10.99
64K	64Kx1	120 ns	2.10
64K	64Kx1	150 ns	1.40
EPROM			
27C256	32Kx8	250 ns	$20.99
27256	32Kx8	250 ns	15.27
27128	16Kx8	250 ns	5.17
27C64	8Kx8	200 ns	8.15
2764	8Kx8	250 ns	3.64
2732A	4Kx8	250 ns	3.95
2716	2Kx8	450 ns	2.95
STATIC RAM			
6264LP-15	8Kx8	150 ns	$8.75
6116LP-3	2Kx8	150 ns	2.50

QUANTITY ONE PRICES SHOWN

OPEN 6½ DAYS: WE CAN SHIP VIA FED-EX ON SAT.

MasterCard VISA or UPS CASH COD
Factory New, Prime Parts µP∞
MICROPROCESSORS UNLIMITED
24,000 S. Peoria Ave.
BEGGS, OK 74421 **(918) 267-4961**

Prices shown above are for April 22, 1985

Please call for current prices. Prices subject to change. Please expect higher or lower prices on some parts due to supply & demand and our changing costs. Shipping & insurance extra. Cash discount prices shown. Orders received by 6 PM CST can usually be delivered to you by the next morning, via Federal Express Standard Air -- $6.00, or Priority One -- $11.50!

Circle **no. 64** on reader service card.

Now With Windowing! $49.95 Basic Compiler

MTBASIC

Features:

Multitasking	Windowing
Handles interrupts	Interactive
Fast native code	Compiles quickly
Floating point	No runtime fee

MTBASIC is a true native code compiler. It runs *Bytes*'s Sept. '81 seive in 26 seconds; interpreters take over 1400 seconds! Because MTBASIC is multitasking, it can run up to 10 Basic routines at the same time, while displaying ten separate windows. Pop-up/down menus are a snap to implement.

MTBASIC combines the best of interpreters and compilers. To the programmer, MTBASIC appears to be an extremely fast interpreter. MTBASIC compiles a typical 100 line Basic program in 1 second, yet it generates blindingly fast code. No more waiting for long compiles.

AVAILABLE for CP/M (Z-80) and PC-DOS systems.

ORDERING: Specify format when ordering. We accept Visa, MC, checks and COD. Send $49.95 plus $3.50 shipping and handling ($10 overseas) to:

SOFTAID, Inc.

P.O. Box 2412 Columbia, MD 21045-1412
301/792-8096

Circle **no. 88** on reader service card.

NEW FEATURES

(Free update for our early customers!)

- Edit & Load multiple memory resident files.
- Complete 8087 assembler mnemonics.
- High level 8087 support.
  Full range transcendentals
  (tan, sin, cos, arctan,
  logs and exponentials)
  Data type conversion and
  I/O formatting.
- High level interrupt support.
  Execute Forth words from within machine code primitives.
- 80186 Assembler extensions for Tandy 2000, etc.
- Video/Graphics interface for Data General Desktop Model 10

HS/FORTH

- Fully Optimized & Tested for:
  IBM-PC IBM-XT IBM-JR
  COMPAQ EAGLE-PC-2
  TANDY 2000 CORONA
  LEADING EDGE
  (Identical version runs on almost all MSDOS compatibles!)
- Graphics & Text
  (including windowed scrolling)
- Music - foreground and background
  includes multi-tasking example
- Includes Forth-79 and Forth-83
- File and/or Screen interfaces
- Segment Management Support
- Full megabyte - programs or data
- Complete Assembler
  (interactive, easy to use & learn)
- Compare

BYTE Sieve Benchmark jan 83
HS/FORTH 47 sec BASIC 2000 sec
w/AUTO-OPT 9 sec Assembler 5 sec
other Forths (mostly 64k) 70-140 sec

FASTEST FORTH SYSTEM AVAILABLE.

TWICE AS FAST AS OTHER FULL MEGABYTE FORTHS!

(TEN TIMES FASTER WHEN USING AUTO-OPT!)

HS/FORTH, complete system only: $250.

VISA Visa Mastercard

Add $10. shipping and handling

HARVARD SOFTWORKS

P.O. Box 69
Springboro, OH 45066
(513) 748-0390

Circle no. 44 on reader service card.

ADVERTISER INDEX

Reader Service No.	Advertiser	Page No.
4	AGSComputers, Inc.	C-2
1	Alpha Computer Service	87
5	Amanuensis, Inc.	55
8	American Planning Corp.	87
3	Anthropomorphic Systems	127
16	Arity Corporation	35
7	Artisoft	113
130	Automata Design Associates	83
12	BD Software, Inc.	28
11	Borland International	2-3
14	Borland International	C-4
*	Business Software Magazine	91
19	C Source	120
17	C User's Group	127
18	C Ware	81
13	CDE Software	29
21	Chalcedony	14
23	Code Works	43
9	CompuView	89
24	CompuServe	C-3
25	Computer Friends	78
22	Computer Helper Industries, Inc.	120
96	Computer Innovations	121
57	Consulair Corp.	29
29	Continuum Microsystems Ltd.	127
32	Corporate Microsystems	23
27	Creative Solutions	95
28	D&W Digital	30
*	*DDJ* Classified Advertising	99
31	Data Base Decisions	73
6	Datalight	97
2	Davong	13
33	Digital Research Computers	49
35	Ecosoft, Inc.	105
114	Emerald Systems	123
36	Essential Software	57
37	Faircom	27
40	Fox Software, Inc.	67
*	Gimpel Software	77
*	Gimpel Software	58
43	Greenleaf Software, Inc.	17
26	Hallock Systems Consultants	73
44	Harvard Softworks	128
68	IBM	116-117
108	IQ Software	109
46	Illyes Systems	84
91	Info Pro Systems	120
53	Inmac	111
*	Integral Quality	86
15	Integrand Research Corp.	87
50	Interface Technologies	39
*	J. D. Owens & Assoc.	105
30	Kenmore Computer Tech.	98
55	Laboratory Microsystems Inc.	105
104	Lahey Computer Systems	88
58	Lattice, Inc.	83
*	MIX Software	115
62	Manx Software	19
84	Megamax, Inc.	43
34	Micro Solutions	12
134	Micro Software Developers	75
41	MicroSmith	127
64	Microprocessors Unlimited	127
38	Microtec	1
66	Mitek	51
128	Morgan Computing	127
39	Motel Computers, Ltd.	127
42	Multi-Tech Electronics	45
44	National Memory Systems	125
76	New Generation Systems	79
70	Phoenix Computer Products	11
69	Plu Perfect Systems	75
71	Poor Persons Software	79
72	Programmer's Shop	59
79	Rational Systems, Inc.	70
80	Revasco	73
78	SLR Systems	97
47	STS Enterprises	127
85	SemiDisk Systems	81
86	Shaw American Technologies	79
63	Soft Advances	91
88	Softaid, Inc.	127
89	Softfocus	98
49	Softronics	119
90	Software Horizons, Inc.	111
61	Software Toolworks	51
92	Softway, Inc.	33
75	Solution Systems	63
93	Solution Systems	63
94	Solution Systems	63
95	Solution Systems	63
102	Solution Systems	47
118	Solutionware	88
65	Spruce Technologies Corp.	90
82	Tatum Labs	51
67	TeamMate	124
81	Turbo Power Software	76
52	US Robotics	7
77	UniPress Software	41
60	Varatek Software Architects	111
112	Wendin, Inc.	9
59	Western Wares	88
116	Wizard Systems	81
51	Wordtech Systems	126
126	Zeducomp	93
*	DDJ Advertising	83
*	DDJ Back Issues	103
*	DDJ Bound Volume	85, 122
*	DDJ C Compiler	97
*	DDJ Subscription	91

Advertising Sales Offices

East Coast
Walter Andrzejewski (617) 567-8361

Midwest/West Central
Michele Beaty (317) 875-0557

Northern California/Northwest
Lisa Boudreau (415) 424-0600

Southern California/Southwest
Beth Dudas (714) 643-9439

Classified Advertising
Alice Abrams (415) 424-0600

Advertising Director
Stephen Friedman (415) 424-0600

Advertising Coordinators
Jay Horvath (415) 424-0600
Alice Abrams (415) 424-0600

SQUEEZE MORE OUT OF EVERY ON-LINE MINUTE.

WITH NEW VIDTEX™ COMMUNICATIONS SOFTWARE FROM COMPUSERVE.

Presenting the software package that makes your computer more productive and cost-efficient.

CompuServe's new Vidtex™ is compatible with many personal computers sold today (including Apple,® Commodore,® and Tandy/Radio Shack® brands). And it offers the following features*–and more–to let you communicate more economically with most time-sharing services (including CompuServe's Information Service).

Auto-Logon. Lets you log on to a host simply and quickly by utilizing prompts and responses defined by you. Also allows quick transmission of predefined responses to host application programs after logging on.

Function Keys. Let you consolidate long commands into single keystrokes. Definitions can be saved to and loaded from disk file, allowing multiple definitions for multiple applications.

Error-Free Uploading and Downloading. CompuServe "B" Protocol contained in Vidtex lets you transfer from your computer to CompuServe and from CompuServe to your computer anywhere in the country. Also provides error-free downloading from CompuServe's extensive software libraries.

Full Printer Support. Printer buffer automatically buffers characters until printer can process; automatically stops on-line transmission when full; and automatically resumes transmission when capacity is re-established. Also, lets you print contents of textual video screen or RAM buffer at any time.

Capture Buffer. Saves selected parts of a session. Contents can be written to a disk file; displayed both on and off line; loaded from disk; and transmitted to the host.

On-line Graphics. Integral graphics protocol displays stock charts, weather maps and more.

If you are already a CompuServe subscriber, you can order Vidtex on line by using the GO ORDER command. Otherwise, check with your nearest computer dealer; or to order direct, call or write:

CompuServe

P.O. Box 20212, 5000 Arlington Centre Blvd.
Columbus, Ohio 43220

1-800-848-8199
In Ohio, call 614-457-0802

An H&R Block Company

*Some versions of the Vidtex software do not implement all features listed.

Vidtex is a trademark of CompuServe, Incorporated. Apple is a trademark registered to Apple Computer, Inc. Commodore is a trademark of Commodore Business Machines. Radio Shack is a trademark of Tandy Corp.

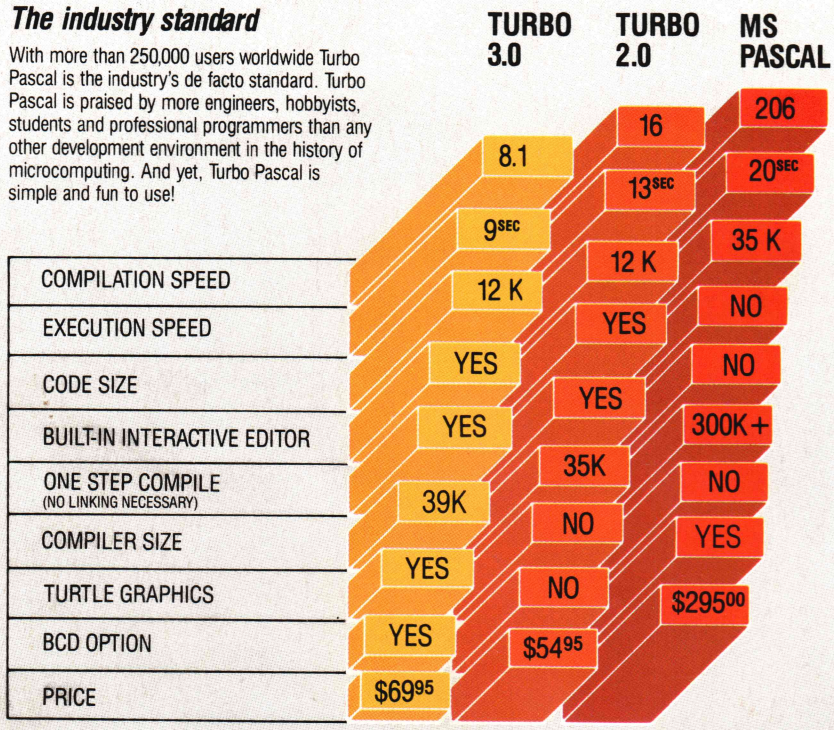
Circle **no. 24** on reader service card.

MOST SIGNIFICANT PRODUCT OF THE YEAR - PC WEEK

They said it couldn't be done. Borland Did It. Turbo Pascal 3.0

The industry standard

With more than 250,000 users worldwide Turbo Pascal is the industry's de facto standard. Turbo Pascal is praised by more engineers, hobbyists, students and professional programmers than any other development environment in the history of microcomputing. And yet, Turbo Pascal is simple and fun to use!



The best just got better: Introducing Turbo Pascal 3.0

We just added a whole range of exciting new features to Turbo Pascal:

- First, the world's fastest Pascal compiler just got faster. Turbo Pascal 3.0 (16 bit version) compiles twice as fast as Turbo Pascal 2.0! No kidding.
- Then, we totally rewrote the file I/O system, and we also now support I/O redirection.
- For the IBM PC versions, we've even added "turtle graphics" and full tree directory support.
- For all 16 Bit versions, we now offer two additional options: 8087 math coprocessor support for intensive calculations and Binary Coded Decimals (BCD) for business applications.
- And much much more.

The Critics' Choice.

Jeff Duntemann, PC Magazine: *"Language deal of the century . . . Turbo Pascal: It introduces a new programming environment and runs like magic."*

Dave Garland, Popular Computing: *"Most Pascal compilers barely fit on a disk, but Turbo Pascal packs an editor, compiler, linker, and run-time library into just 39K bytes of random-access memory."*

Jerry Pournelle, BYTE: *"What I think the computer industry is headed for: well documented, standard, plenty of good features, and a reasonable price."*

Portability.

Turbo Pascal is available today for most computers running PC DOS, MS DOS, CP/M 80 or CP/M 86. A XENIX version of Turbo Pascal will soon be announced, and before the end of the year, Turbo Pascal will be running on most 68000 based microcomputers.

(*) Benchmark run on an IBM PC using MS Pascal version 3.2 and the DOS linker version 2.6. The 179 line program used is the "Gauss-Seidel" program out of Alan R. Miller's book: *Pascal programs for scientists and engineers* (Sybex, page 128) with a 3 dimensional non-singular matrix and a relaxation coefficient of 1.0.

An Offer You Can't Refuse.

Until June 1st, 1985, you can get Turbo Pascal 3.0 for only $69.95. Turbo Pascal 3.0, equipped with either the BCD or 8087 options, is available for an additional $39.95 or Turbo Pascal 3.0 with both options for only $124.95. As a matter of fact, if you own a 16-Bit computer and are serious about programming, you might as well get both options right away and save almost $25.

Update policy.

As always, our first commitment is to our customers. You built Borland and we will always honor your support.

So, to make your upgrade to the exciting new version of Turbo Pascal 3.0 easy, we will accept your original Turbo Pascal *disk* (in a bend-proof container) for a trade-in credit of $39.95 and your Turbo87 original *disk* for $59.95. This trade-in credit may only be applied toward the purchase of Turbo Pascal 3.0 and its additional BCD and 8087 options (trade-in offer is only valid directly through Borland and until June 1st. 1985).

NOT COPY-PROTECTED

TURBO PASCAL

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133

Carefully Describe your Computer System!

Mine is: ___ 8 bit ___ 16 bit
I Use: ___ PC-DOS ___ MS-DOS
___ CP/M 80 ___ CP/M 86

My computers' name/model is: _____

The disk size I use is:
☐ 3½" ☐ 5¼" ☐ 8"

Name: _____

Shipping Address: _____

City: _____ Zip: _____

State: _____

Telephone: _____

For update: original Turbo disk must accompany order

YES! I want the Best!

Please send:		Quantity
Pascal 3.0	$ 69.95	
Pascal w/8087	$109.90	
Pascal w/BCD	$109.90	
Pascal w/8087 & BCD	$124.95 (SAVE $24.90)	

• These prices include shipping to all U.S. cities. All foreign orders add $10 per product ordered.

Subtotal (CA 6% tax) _____
Trade-in Credit Claimed: _____
Amount Enclosed: _____

Payment: VISA MC BankDraft Check

Credit Card Expir. Date: ___/___

Card #: _____

COD's and Purchase Orders WILL NOT be accepted by Borland. California residents: add 6% sales tax. Outside USA: add $10 and make payment by bank draft, payable in US dollars drawn on a US bank.

11

BORLAND INTERNATIONAL

Software's Newest Direction
4585 Scotts Valley Drive
Scotts Valley, CA 95066
TELEX 172373

Turbo Pascal is a registered trademark of Borland International, Inc.
PC Week is a trademark of Ziff-Davis Pub. Co.

Circle **no. 14** on reader service card.